AD 407 958

Coordinated Science Laboratory

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

ON DIRECT FIXED-TIME OPTIMIZATION OF INVERTIBLE SYSTEMS

Stephen James Kahne

REPORT R-162 **JUNE, 1963**

COORDINATED SCIENCE LABORATORY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

Contract DA-36-039-TR US AMC 02208(E)
DA Project 3A-99-25-004

The research reported in this document was made possible by support extended to the University of Illinois, Coordinated Science Laboratory, jointly by the Department of the Army, Department of the Navy (Office of Naval Research), and the Department of the Air Force (Office of Scientific Research) under Department of Army Contract DA-36-039-TR US AMC 02208(E).

ABSTRACT

A numerical technique for calculating the optimal control for a class of systems and constraints is described. Nonlinear, time-varying deterministic systems subject to hard state space and hard control space constraints are considered. Three numerical procedures are developed to perform the optimization. A technique for the minimization of a scalar function of a vector variable is described where the components of the vector are constrained by upper and lower bounds. This minimization procedure is incorporated in a method of constraint mapping which maps the state space constraints into the control space. To improve convergence properties of the optimization procedure the notion of a pseudo performance index is introduced. Initial and final states may be partially or completely specified. Any unspecified initial or final state vector components are optimally selected.

An iterative technique for the optimization is demonstrated which generally converges to a local minimum of the performance index. The method uses the direct approach to optimization and is very efficient computationally. Examples of space vehicle trajectory optimization problems are given.

ACKNOWLEDGMENT

The author is indebted to his advisor, Dr. J. B. Cruz, Jr., for advice and encouragement during the period of this research.

He is grateful for the financial support of the Coordinated Science Laboratory, Department of Electrical Engineering and the Ford Foundation. The staff and facilities of the Coordinated Science Laboratory have contributed significantly to the successful completion of this study.

CONTENTS

1. PERSPECTIVE - I

1.1 Recent Trends in Modern System Theory

For the purpose of this work, we will consider that modern system theory has evolved from the 1950's when the state variable formulation of control problems was introduced and studied by Bellman[1]. Of course we recognize that the state variable formalism is not new; only its application to control problems is recent. This is a convenient, if somewhat artificial, beginning.

As noted elsewhere[2,3] the use of the modern theory enables the designer to consider his problem in the time domain with the physical time domain constraints as an integral part of the problem formulation. The so-called "cut-and-try" techniques of the classical control theory are no longer necessary. However, a certain new type of "cut-and-try" design is introduced. This new scheme is systematized to the extent that algorithms are created which assure that each succeeding try is better than the previous one, a process which we call monotone iteration. The most significant property of monotone iterative procedures is that by using them we are always assured of doing better if we persist. That is, if n iterations have produced a certain result, then the (n + 1)st iteration will produce a "better" result. The method of solution to be described in Chapter 3 is an example of such monotone iteration.

The term "better" is given a precise meaning by the introduction of a mathematical performance index. This is a functional which represents how well the system is doing. Generally we ask for this functional to be taken to an extreme value. Without loss of generality, we will henceforth assume that it is to be minimized. Now, in many systems an appropriate input (a time function) can be generated which will drive the system through its state space (in time) in such a manner that the performance index is indeed minimized. However, in many systems, such control (i.e., such an input) would also lead to gross misbehavior of the system. This may arise from saturation nonlinearities, oscillatory modes of operation, excessive fuel requirements, excessive aerodynamic pressures, velocity, etc. It is for such reasons that many physically meaningful problems involve constraints on the state of the system and the inputs to the system. We say that, in order to avoid certain undesirable behavior of the system, state space constraints and control space constraints are imposed.

The evolution of modern system theory has passed through many different phases of development during which time a variety of systems and constraints have been considered. It is hardly necessary here to discuss these systems and constraints which have been considered in the past. It is sufficient to point out that the work described in these pages is the subject of a large amount of current effort by system theorists and mathematicians.

The work described in these pages is an engineering solution to the optimal control problem for totally constrained (i.e., state and control space hard constraints) invertible systems. The constraints considered are practical ones. The method of solution is computationally feasible and completely automatic in that, no decisions by the computer operator are required during the course of the computation.

Section 1.2 contrasts this method of solution with others recently proposed.

1.2 State Space Constraints

In the evolution of modern system theory mentioned above the first constraints to be considered were control space constraints. These typically required the control vector to be containedin a closed region of the control space. Until 1959 there was no mention of state space constraints. This was probably true for two reasons. The control space constraints provided sufficiently complicated analytical difficulties by themselves and also, there were not sufficient results published to attract attention to certain analytical results which proved to be physically unrealizable. In 1959, a fundamental paper by Gamkrelidze* considered the problem of state variable constraints in some detail. His paper delineates a set of necessary conditions which a system trajectory and control must satisfy in order to be optimal when the state of the system is restricted to lie in a closed region and the control is also restricted to lie in a closed region. It is shown that those portions of the optimal trajectory which fall entirely within the closed region in the state space must satisfy the maximum principle. A necessary condition is also proven for the portions of the trajectory which lie entirely on the boundary of the closed region in the state space. Further, a jump condition is defined which is necessarily satisfied by every pair of adjoining sections of an optimal trajectory, one of which lies in the interior and the other on the boundary of this closed region. These results give certain analytical properties of optimal trajectories but do not seem to yield easily to computational solution of optimal programming problems. Berkovitz[26] obtained the same results using the calculus of variations.

In 1961, Chang[5] determined a simpler set of necessary conditions for the special case of fixed time optimal control with free end point. Moreover, he showed that for linear systems, if the restricted regions in the state space and in the control space are convex, the condition stated is also sufficient. This condition also holds for minimal time control between two fixed points. Again, the main emphasis in Chang's paper is to derive necessary (and sometimes sufficient) conditions with little regard for computational feasibility.

---

*Recently rewritten[4]

Some consideration of the computational aspects of optimal programming problems was given in Breakwell's[6] paper in 1959. However, in 1961, several papers appeared which seriously considered the computational aspects of the optimization problem. Some of these did not consider any state space or control space constraints[7,8]; some considered only control space constraints[9]; some considered only state space constraints[10]; and some considered both state and control space constraints[11]. In July 1962, Dreyfus published a paper[12] which summarized his earlier work and gave some numerical results and discussion of a computational technique for the totally constrained problem. He was interested in a single, time independent constraint on the state variables and a scalar control variable of a more general type than Gamkrelidze considered. Dreyfus derives the computational procedure through use of a dynamic programming formulation of the problem. In this way he obtains expressions for incremental improvements in the control program at each step in the iteration.

Another scheme which has gained wide acceptance falls in the general category of gradient techniques. Kelly[7,13,14] was an early advocate of gradient methods applied to optimal programming problems.

The two most significant results dealing with the topic of the present work are due to Ho and Brentani[15] and Denham and Bryson[16] both of which appeared in November 1962. In the paper by Ho and Brentani[15], fixed time problems are considered with hard inequality constraints either in the state space or the control space. They note that considerable difficulty is encountered for non-linear systems using their method. Also, the restriction to either state or control space constraints is a significant one. On the other hand, Denham and Bryson[16], considered general state and control space constraints with a single inequality constraint and a scalar control variable. Their method is essentially one of steepest descent. A striking disadvantage of their approach is that complicated on-line calculations and decisions by the researcher are required during computation[17]. They consider free end points and free final time problems as well as those mentioned above.

It is well to note exactly the class of problems being solved in the present work in order to place it among the efforts mentioned above. Here we consider non-linear, time-varying, deterministic systems which are invertible (to be defined later). The constraints are simultaneously imposed in the state and control spaces and are hard inequality constraints. The performance index is a general function of the state and the control over a prescribed time interval. In general, both the state and control variables will be vectors. The technique to be described is a gradient-type method based on two principles. First, a mapping is defined which collects all state and control space constraints into the control space*. Secondly, a minimization technique[18] is used to successively select controls which reduce the performance index.

It would appear that this method and that of Denham and Bryson are competitive and should, in the future, be carefully compared. The method of this work is direct**, whereas most of the other authors have concentrated on indirect techniques. In the opinion of the author, there has not been conclusive evidence that one scheme is better than the other for all problems. Even so, only little work has been done on direct methods of solution. It seems that, at this stage of technology, we cannot exclude either approach.

A recent publication of Friedland[28] gives some evidence that several computational techniques for nonlinear programming are currently being compared in a unified study. It is felt that work such as this is necessary to effectively use appropriate techniques for specific problems.

*This is a generalization of a result published earlier by the author[3].

**A direct method depends on successive comparisons of a function. An indirect method seeks a minimum by means of a necessary condition for the minimum[14].

2. THE PROBLEM

In what follows, capital Roman letters will denote matrices whose dimensions will be obvious from the text. Small underlined Roman letters will denote vectors. Again, dimensions will be obvious. Any deviations from this notation will be clearly indicated in the text.

An expression involving functions and functionals will be accompanied by a statement indicating its domain of definition. For example:

$$\dot{\underline{x}}(t) = f[x(t)]; \quad t \in (0,T] \tag{2.0.1}$$

is an equation relating a vector function $\dot{\underline{x}}(t)$ and a vector functional $\underline{f}$ defined on $\underline{x}(t)$ for $t \in (0,T]$ where (and] have their usual meanings. The equation

$$\dot{\underline{x}}(t) = f[x(t)] \tag{2.0.2}$$

is an expression involving vectors of numbers. For the discrete case, a similar distinction is made

$$\underline{x}(k + \Delta) = \underline{f}[\underline{x}(k), \underline{u}(k), \Delta]; \quad k = 1, 2, \ldots, K \tag{2.0.3}$$

and

$$\underline{x}(k + \Delta) = \underline{f}[\underline{x}(k), \underline{u}(k), \Delta] \tag{2.0.4}$$

Only stationary systems will be considered since time varying systems of equations can be written as stationary ones by introduction of another state variable[2].

2.1 Definition of Invertible Systems

If

$$\underline{\dot{x}}(t) = \underline{f}[\underline{x}(t), \underline{u}(t)]; \; t \in [0,T] \tag{2.1.1'}$$

we say that (2.1.1') is invertible if there exists a single valued vector function $\underline{g}$ such that

$$\underline{u}(t) = \underline{g}[\underline{\dot{x}}(t), \underline{x}(t)]; \; t \in [0,T] \tag{2.1.2'}$$

Since we are dealing with a problem of numerical solution of optimization problems, we immediately turn our attention to the discrete version of (2.1.1') and (2.1.2'), namely

$$\underline{x}(k + \Delta) = \underline{f}[\underline{x}(k), \underline{u}(k), \Delta]; \; k = 1, 2, \ldots, K \tag{2.1.1}$$

and

$$\underline{u}(k) = \underline{g}[\underline{x}(k + \Delta), \underline{x}(k), \Delta]; \; k = 1, 2, \ldots, K \tag{2.1.2}$$

Two general classes of invertible systems may be defined as follows.

1. Class α invertible systems.

This class is made up of two subclasses:

a) Class α_1 (linear control) invertible systems:

$$\underline{x}(k + \Delta) = \underline{f}[\underline{x}(k), \Delta] + B[\underline{x}(k), \Delta] \, \underline{u}(k); \; k = 1, 2, \ldots, K \tag{2.1.3}$$

where, $\underline{x}$ is an n-vector and $\underline{u}$ is an m-vector, B is an n x m matrix with the typical element

$$b_{ij} = b_{ij}[\underline{x}(k), \Delta]; \; k = 1, 2, \ldots, K \tag{2.1.4}$$

Moreover, it is required that the matrix B have rank m. This requirement is necessary to insure the independence of the control variables.

b) Class α_2 (non-linear, diagonal control) invertible systems:

$$\underline{x}(k + \Delta) = \underline{f}[\underline{x}(k), \Delta] + B[\underline{x}(k), \Delta] \, H[\underline{u}(k)]; \; k = 1, 2, \ldots, K \tag{2.1.5}$$

where B is as described for class α_1 invertible systems and H is an m x m diagonal matrix with the following typical elements each of which has an inverse:

$$h_{ij} = \begin{cases} h_{ii}\,[u_i(k)] & i = j \\ 0 & i \neq j \end{cases} ; \; k = 1, 2, \ldots, K \tag{2.1.6}$$

2. Class β invertible systems:

This class contains all invertible systems not contained in class α.

The development to follow will explain in detail a procedure for optimization of systems which are class α_1 invertible. The results are immediately extended to class α_2 and β invertible systems.

2.2 Statement of the Problem

Given the following class α_1 invertible system

$$\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t)] + B[\underline{x}(t)] \, \underline{u}(t) \tag{2.2.1}$$

determine

$$\underline{u}(t)$$

such that

$$F[\underline{x}(t),\ \underline{u}(t),\ T] \tag{2.2.2}$$

is minimized and

$$\underline{x}^{-}(t) \leq \underline{x}(t) \leq \underline{x}^{+}(t) \tag{2.2.3}$$

and

$$\underline{u}^{-}(t) \leq \underline{u}(t) \leq \underline{u}^{+}(t) \tag{2.2.4}$$

all for $t \in [0,T]$. Statements such as (2.2.3) signify a component by component relationship between vectors.

We will assume that artificial constraints will be imposed in the absence of any components of (2.2.3) or (2.2.4). These will be of the form

$$-\underline{L} \leq \underline{x}(t) \leq +\underline{L};\ -\underline{L} \leq \underline{u}(t) \leq +\underline{L},$$

where $\underline{L}$ is a vector of suitably large positive numbers.

The terms of this problem statement are defined below for $t \in [0,T]$.

$\underline{x}(t)$	(n-dimensional) state vector
$\underline{x}^{\pm}(t)$	(n-dimensional) state constraint vectors
$\underline{u}(t)$	(m-dimensional) control vector
$\underline{u}^{\pm}(t)$	(m-dimensional) control constraint vectors
$\underline{f}$	(n-dimensional) system vector
B	(n x m dimensional) interaction matrix
F	performance functional

A typical example of the performance functional is

$$F = \int_0^T [a \sum_{i=1}^{n} x_i^r(t) + b \sum_{i=1}^{m} u_i^s(t)]\, dt + p \sum_{i=1}^{n} |x_i(T)|$$

The control vector $\underline{u}^o(t)$; $t \in [0,T]$, which satisfies (2.2.2) subject to (2.2.1), (2.2.3), (2.2.4) is called the globally optimum control vector. If $\underline{u}'(t)$; $t \in [0,T]$, is such that, for all $\delta\underline{u}(t)$; $t \in [0,T]$, $F[\underline{x}'(t), \underline{u}'(t), T] < F[\underline{x}'(t) + \delta\underline{x}(t), \underline{u}'(t) + \delta\underline{u}(t), T]$; $t \in [0,T]$, then $\underline{u}'(t)$ is a locally optimum control vector. Note that the perturbed controls and states must satisfy (2.2.3) and (2.2.4) and $\delta\underline{u}(t)$ are small.

The techniques of this study determine a locally optimum control vector. To find the globally optimum control vector, we simply select the 'best' locally optimum one. By this we mean that if ${}^1\underline{u}(t), {}^2\underline{u}(t), \ldots, {}^k\underline{u}(t)$; $t \in [0,T]$ are locally optimal, then there exists a j such that, for all $i \neq j$,

$$F[{}^j\underline{x}(t), {}^j\underline{u}(t), T] < F[{}^i\underline{x}(t), {}^i\underline{u}(t), T]; \; t \in [0,T].$$

We call ${}^{j}\underline{u}(t)$; $t \in [0,T]$ the globally optimal control vector. Since the number of locally optimal control vectors may be very large, generally, as in this work, one is satisfied to find a locally optimal control vector. Moreover, in many problems, physical reasoning shows that there is only one locally optimal control vector and therefore it is globally optimal.

3. A SOLUTION

3.1 Introduction to Solution

The method of solution to be described has two main subdivisions. First, the state space constraints at time k + 1 are mapped into the control space at time k. These constraints along with the control space constraints at time k, determine the subspace of the control space which contains all admissible sets of controls at time k. This is the constraint mapping procedure. Next, the control at time k is selected which minimizes a pseudo performance index. At the end of a major iteration it will be shown that no control or state space constraints are violated and that the performance index originally specified (in (2.2.2)) is strictly smaller than it was at the end of the previous major iteration.

The present chapter contains the details of the method outlined above. First we consider the constraint mapping procedure in some detail, then discuss the particularly important one step map which will prove useful later. Next the method of adaptive constrained descent is described. This technique is used both for the mapping and later for the optimization. A brief discussion of the augmented performance index follows. The chapter concludes with a detailed description of the overall optimization procedure using the schemes described earlier in the chapter.

As remarked earlier, this work is especially written for class α_1 invertible systems as explained in Chapter Two.

3.2 Constraint Mapping

It is convenient to define several sets in the space of real numbers. The notation follows.

In the state space, $X^j_{k_i}$ is the set of all state variables $x_j(k)$ at time k_i. $X^j_{k_i+1}$ is the set of all state variables $x_j(k)$ at time $k_i + 1$. Of course, the set of all $x_j(k)$ may be ordered and assembled into a state vector $\underline{x}(k)$ at time k. The underline signifies a vector whose components are $x_j(k)$.

In the control space we have the following sets to be defined below: ${}^1U_k^j$, ${}^2U_k^j$, ${}^3U_k^j$, ${}^*U_k^j$, ${}^{**}U_k^j$. In general the subscript indicates the time at which the set is defined (i.e., ${}^*U_1^j$ is the set ${}^*U^j$ at time k = 1). Thus, for example, ${}^*u_j(k) \in {}^*U_k^j$ is the j^{th} element of a particular control vector ${}^*\underline{u}(k)$ at time k. ϕ is the null set. R is the real line. A bar over a set (e.g., $\overline{{}^*U_k^j}$) denotes its complement. The sets are defined in terms of the problem statement of (2.2.1), (2.2.2), (2.2.3), and (2.2.4) for j = 1, 2, ..., m.

The technique described below shows how to construct a set ${}^1U_k^j$ so that ${}^1U_k^j = \overline{{}^3U_k^j}$.

$u_j(k) \in {}^3U_k^j$ implies violation of (2.2.1), (2.2.3). A control variable selected from ${}^3U_k^j$ is sufficient to drive the system outside the allowable region in the state space (at time k + 1).

$u_j(k) \in {}^2U_k^j$ implies satisfaction of (2.2.4). The set so defined is the connected set on R whose boundaries are $u_j^+(k)$ and $u_j^-(k)$ in (2.2.4).

${}^{1'}U_k^j \subseteq {}^1U_k^j$

$u_j(k) \in {}^{1'}U_k^j$ implies satisfaction of (2.2.1), (2.2.3). A control variable selected from ${}^{1'}U_k^j$ necessarily drives the system into an allowable state at time k + 1, where the allowable states are defined by (2.2.3).

$^{*}U_k^j = {}^{1}U_k^j \bigcap {}^{2}U_k^j$

$u_j(k) \notin {}^{*}U_k^j$ implies violation of (2.2.1), (2.2.3), and/or (2.2.4). In order to drive the system into an allowable (defined by (2.2.3)) state at time k + 1, it is necessary to select a control from ${}^{*}U_k^j$.

$^{**}U_k^j = {}^{1'}U_k^j \bigcap {}^{2}U_k^j$

$u_j(k) \in {}^{**}U_k^j$ implies satisfaction of (2.2.1), (2.2.3), (2.2.4). In order to drive the system into an allowable state at time k + 1, it is necessary and sufficient to select a control from ${}^{**}U_k^j$.

It is clear that all these sets are compact.

The set ${}^{**}U_k^j$ is the set of all controls which are candidates for the "optimal" control. It is this set which we would like to calculate. It will be shown below that ${}^{**}U_k^j$ is a function of the state and the control at time k. The conjecture is made that for general systems, or even for general invertible systems ${}^{**}U_k^j$, cannot be calculated. An example is given which shows how to calculate ${}^{**}U_k^j$ in a special case. A set ${}^{*}U_k^j$, the construction of which will be explained, is a function of the present state of the system and, in general, appears to be the smallest set which can be easily calculated which contains the optimal control. This question of a "best" ${}^{*}U_k^j$ is a subject which deserves further attention. It is not discussed in this work.

The calculation of ${}^{*}U_k^j$, called a one step map is easily performed for the class α_1 invertible systems by determining its endpoints ${}^{*}u_j^{\pm}(k)$. This, it will be shown, only requires examination of $x_j^{\pm}(k + 1)$. For general invertible systems the space X_{k+1} must be searched.

The calculation of ${}^{*}U_{k_i}^{j}$ proceeds as follows:

1. Calculate ${}^{1}U_{k_i}^{j}$: Given $\underline{x}(k_i)$, search $x_j(k_i + 1)$ such that $x_j^{-}(k_i + 1) \leq x_j(k_i + 1) \leq x_j^{+}(k_i + 1)$ until the $u_j(k_i)$ of the inverted system reaches its maximum and minimum values. These boundaries define a compact, connected set ${}^{1}U_{k_i}^{j}$ on R.

2. Calculate ${}^{2}U_{k_i}^{j}$: $u_j^{+}(k_i)$ and $u_j^{-}(k_i)$ are the end points of ${}^{2}U_{k_i}^{j}$ on R.

3. Calculate ${}^{*}U_{k_i}^{j}$: ${}^{*}U_{k_i}^{j} = {}^{1}U_{k_i}^{j} \bigcap {}^{2}U_{k_i}^{j}$.

In general, the searching operation of step 1 above will require an iterative numerical procedure. However, if the inverted system can be represented by

$$u_i(k) = x_i(k + 1) - g_i(\underline{x}(k)) \qquad \begin{array}{l} i = 1, 2, \ldots, m \\ k = 1, 2, \ldots, K \end{array} \tag{3.2.1}$$

then $u_i^{+}(k)$ will occur when $x_i(k + 1) = x_i^{+}(k + 1)$ and $u_i^{-}(k)$ will occur when $x_i(k + 1) = x_i^{-}(k + 1)$ so the searching procedure is unnecessary for the case of (3.2.1). It might be noted that most systems can be represented as in (3.2.1).

For the general case, a numerical procedure has been developed to perform the necessary boundary mapping. A description of this procedure, the method of adaptive constrained descent is given in the next section.

In light of the preceding discussion of constraint mapping, let us carefully examine its application to class α_1 invertible systems. It is shown later that we are interested in the one step map problem which yields the set ${}^{*}U_k^{j}$. For the α_1-type systems

$$\underline{x}[k + \Delta] = \underline{f}[\underline{x}(k), \Delta] + B[\underline{x}(k), \Delta]\, \underline{u}(k) \tag{3.2.2}$$

For a given state $\underline{x}(k)$ we can write (3.3.2) as

$$\underline{x}[k + \Delta] = \underline{f}[\underline{x}(k), \Delta] + \underline{v}(k) \tag{3.2.3}$$

where

$$\underline{v}(k) = B[\underline{x}(k), \Delta]\, \underline{u}(k) \tag{3.2.4}$$

For independent controls $u_i(k)$ this transformation always has an inverse and thus $\underline{u}(k)$ can be determined from $\underline{v}(k)$. Equation (3.2.3) is precisely the form desired for the one step mapping.

For α_2-type systems

$$\underline{x}[k + \Delta] = \underline{f}[\underline{x}(k), \Delta] + B[\underline{x}(k), \Delta]\, H[\underline{u}(k)] \tag{3.2.5}$$

where H is the diagonal matrix described earlier. The requirement here, for the successful application of the one step map technique is that $h_{ii}[u_i(k)]$ have an inverse. The notion of linear independence is not present here and it is difficult to formulate a set of physical requirements on a system for this condition to exist.

For β-type systems, again, it is difficult, if not impossible, to discuss physical interpretations of the invertible nature of non-linear control systems. A thorough study of this problem may very well lead to useful results. For the purpose of the present study, however, such problems are not considered.

This section is concluded by a discussion of the existence of ${}^{**}U_k^j$. It was observed earlier that we could find a ${}^{*}U_k^j$ which is a function of $\underline{x}(k)$. If there were a ${}^{**}U_k^j$ then it would be a function of $\underline{x}(k)$ and $\underline{u}(k)$. To show this, consider the following general inverted system

$$
\begin{aligned}
u_1(k) &= x_1(k + 1) - g_1[\underline{x}(k)] \\
u_2(k) &= x_2(k + 1) - g_2[\underline{x}(k)] \\
&\vdots \\
u_m(k) &= x_m(k + 1) - g_m[\underline{x}(k)]
\end{aligned}
\qquad (3.2.6)
$$

If we selected ${}^{**}u_1^{\pm}(k)$ only on the basis of $x_1^{\pm}(k + 1)$ and $\underline{x}(k)$, it is not clear that this would ensure that $\underline{x}^-(k + 1) \leq \underline{x}(k + 1) \leq \underline{x}^+(k + 1)$. Counterexamples are easily found to show that, in fact, ${}^{**}u_1^{\pm}(k)$ must be selected on the basis of more information than $x_1^{\pm}(k + 1)$. Let us suppose that because of coupling between equations of (3.2.6) the selection of ${}^{**}u_1^{\pm}(k)$ depends on $x_2^{\pm}(k + 1)$ as well as $x_1^{\pm}(k + 1)$. Clearly $x_2(k + 1)$ depends on $u_2(k)$ also. Therefore, it is easily seen that ${}^{**}u_1^{\pm}(k)$ is a function of $\underline{x}(k)$ and $\underline{u}(k)$. However, $\underline{u}(k)$ is as yet unknown. For each $\underline{u}(k)$, ${}^{**}u_1^{\pm}(k)$ will change (as will the other ${}^{**}u_i^{\pm}(k)$). So, in general, it is impossible to find ${}^{**}u_1^{\pm}(k)$ and therefore, for computational purposes, it does not exist. There are special cases for which ${}^{**}U_k^j$ does exist and is equal to ${}^{*}U_k^j$ as determined by the construction described earlier.

Consider a linear system represented as shown below:

$$
\underline{x}(k + 1) = A(k)\, \underline{x}(k) + \underline{U}(k);\ k = 1, 2, \ldots, K \qquad (3.2.7)
$$

where

$$A(k) = \begin{bmatrix} 0 & 1 & 0 - - - - 0 \\ 0 & 0 & 1 - - - - 0 \\ 0 - - - - & - - - - - - & - - - - 1 \\ a_{n1}(k) & a_{n2}(k) - & - - - - - a_{nn}(k) \end{bmatrix}$$

$$\underline{U}^T(k) = [0 \;\; 0 - - - - 0 \;\; u(k)]$$

The system (3.2.7) has a single state variable constraint $x_n^-(k) \leq x_n(k) \leq x_n^+(k)$ and a constraint on the single control $u^-(k) \leq u(k) \leq u^+(k)$. Because the only state variable constraint is on the one variable with a direct input, and since there are no other inputs, no selection of U(k) can drive the variables $x_i(k + 1)$, $i \neq n$ into inadmissible regions (for inadmissible regions for these variables do not exist). Therefore, the procedure for determining ${}^{*}U_k^j$ will give ${}^{**}U_k^j$. This is a special case, to be sure. However, it does show an interesting property of systems which correspond to all-pole systems in the continuous, stationary case. This is not the most general such system, but is satisfactory for the purpose of exhibiting a system for which the ${}^{*}U_k^j$ which we know how to calculate is identical to ${}^{**}U_k^j$.

The fact that ${}^{**}U_k^j$ does not exist in general, enhances the value of ${}^{*}U_k^j$. If we consider the optimization process to be a search of admissible controls, the existence and use of ${}^{*}U_k^j$ reduces the space which must be searched.

3.3 Adaptive Constrained Descent

The method of adaptive constrained descent is one of the so-called direct methods of minimization. The problem is to determine the minimum $Z^i = f(\underline{x}^i)$

such that $\underline{x}^- \leq \underline{x}^i \leq \underline{x}^+$ where f is a scalar function of its vector argument. We let

$$Z^o = \min_{\underline{x}^- \leq \underline{x}^o \leq \underline{x}^+} Z^i = f(\underline{x}^o) \ .$$

The technique generates a set of $\underline{x}^i$ such that $Z^i > Z^{i+1}$. Thus the sequence of Z^i is monotonically decreasing. If, for i = P the technique cannot produce $Z^{P+1} < Z^P$, then $Z^P = Z^o$ and $\underline{x}^P = \underline{x}^o$, the minimizing value of $\underline{x}$ or at least a saddle point of $f(\underline{x})$.

The minimization proceeds down one variable at a time within the restricted region. Each variable is changed until the sensitivity of Z with respect to that variable has been sufficiently reduced. The amount by which the j-th variable is changed at each step is ϵ_j. Initially ϵ_j is set to an appropriate value. Experience shows that for many problems $\epsilon_j = (x_j^+ - x_j^-)/3$ is a reasonable value to start with. The minimization on one variable is continued until the sensitivity of Z with respect to this variable is sufficiently reduced. During this minimization on one (the j-th) variable, ϵ_j is reduced to permit partial covergence to the minimum in one variable. When the sensitivity of Z with respect to this variable is reduced below a cut-off value (minor cycle sensitivity criterion) a minor cycle is complete. If $\underline{x}$ is an n-vector, n minor cycles constitute a major cycle.

The details of the method may be found in Appendix A and elsewhere[18].

3.4 A Pseudo Performance Index

For numerical purposes, it may be helpful to introduce an augmented performance index to account for state space violations which occur at "future" times. This idea will become clear later. We introduce the augmented performance index $H[\underline{x}(k), \underline{u}(k), k]$ defined as the "cost" of going from $\underline{x}(k)$

to $\underline{x}(K)$.

$$H[\underline{x}(k),\ \underline{u}(k),\ k] = F[\underline{x}(k),\ \underline{u}(k),\ K] + G[\underline{x}(k)]$$

$G[\underline{x}(k)]$ is the penalty incurred for violating the state space constraints. (We will insure that $G[\underline{x}(k)] = 0$ at the conclusion of the iterative process.) A convenient form for $G[\underline{x}(k)]$ is

$$G[x(\bar{k})] = \sum_{i=1}^{n} \sum_{k=\bar{k}}^{K} \lambda_{ik}\left\{\delta_{-1}[x_i(k) - x_i^+(k)] + \delta_{-1}[x_i^-(k) - x_i(k)]\right\}\left\{x_i(k) - x_i^*(k)\right\}^2$$

where

$$x_i^*(k) = \frac{x_i^-(k) + x_i^+(k)}{2}$$

and

$$\delta_{-1}(a) = \begin{cases} 0 & a \leq 0 \\ 1 & a > 0 \end{cases}$$

Unless otherwise mentioned we use $\lambda_{ik} = 1$. The minimum of F which is selected may depend on λ_{ik}.

Using the one step mapping procedure we can ensure that the state space constraints at one instant will not be violated. $G[\underline{x}(k)]$ will account for those at later instants which are violated. At any particular time k_i, we can be sure that there are no state or control space constraint violations for $k \leq k_i$, and at $k_i = K$, we can be sure there are no state space constraints violated at any time.

3.5 Direct Fixed-Time Optimization of Invertible Systems

Let us assume that we have no prior knowledge of the optimal control and that, as an alternative to prior knowledge, we have selected $u_i(k) = \mu$; $i = 1, 2, \ldots, m$ and $k = 1, 2, \ldots, K$. Moreover, with this "control", the value of the performance index is ${}^oH = {}^oF + {}^oG$ where H and G were defined in the previous section. F was defined in (2.2.2). The optimization procedure is made up of minor iterations and major iterations. If there are K time instants between $k = 1$ and $k = K$, then one starting iteration and K minor iterations comprise one major iteration.

The starting iteration is the search for "best" initial conditions on the state variables. It is simply a search, using the adaptive constrained descent procedure, over the set of starting values whose boundaries are $\underline{x}^{\pm}(1)$ and a selection of the one which minimizes the augmented performance index. The details of the search are the same as for any of the control vectors, for example $\underline{u}(1)$, and are given below.

The minor iteration has several parts. Upon entering a minor iteration at time k, the following information is known: $\underline{x}(k)$, $\underline{x}^{\pm}(k + 1)$, $\underline{u}^{\pm}(k)$, $H[\underline{x}(k), \underline{u}(k), k]$. Moreover we have constructed the control up to time k so that $G[\underline{x}(1)] = G[\underline{x}(k)]$, which means that there are no state space constraint violations before time k. With this information the iteration proceeds.

1. $i \doteq 1$
2. Use the one step mapping technique to determine ${}^*U_k^i$.
3. Select a $u_i(k) \in {}^*U_k^i$.
4. Check to see if $\underline{x}^-(k + 1) < \underline{x}(k + 1) < \underline{x}^+(k + 1)$. That is, check to see if $u_i(k) \in {}^{**}U_k^i$. If this is true go to 5. If not, go to 3.

5. Compute $H[\underline{x}(k), \underline{u}(k), k]$. Continue selecting $u_i(k) \in {}^{**}U_k^i$ until $H[\underline{x}(k), \underline{u}(k), k]$ is minimized (using the method of Section 3.3).

6. If $i \neq n$, increase i by one and go to 2. If $i = n$, keep a record of the current $H[\underline{x}(k), \underline{u}(k), k]$ and compare it with previous values. If no significant decrease has occurred, this completes the minor iteration. If a significant decrease has occurred, go to 1.

Thus, at the end of a minor iteration at time k, $\underline{u}(k)$ has been set so that each $u_i(k) \in {}^{*}U_k^i$, no state space constraints at time k + 1 are violated (therefore, all $u_i(k) \in {}^{**}U_k^i$; $i = 1, 2, \ldots, m$), and $\underline{u}(k)$ is, in some sense, "best" to date. Moreover, at the end of the minor iteration the states and controls for all previous instants have been calculated and stored. Also, the portion of $H[\underline{x}(1), \underline{u}(1), 1]$ which depends on these states and controls has been calculated and stored. Thus, the calculation time is shortened as the calculation proceeds.

A major iteration consists of one starting iteration and K minor iterations and thus represents a complete pass through the controls $\underline{u}(k)$, $k = 1, 2, \ldots, K$. (See Appendix B.)

A discussion of the stability of the method proposed above and its ability to converge to a critical point of F is given here. First we show a source of instability of this technique. Suppose that $F = {}^{o}F$ at the end of a major iteration. The next minor iteration will select a set of vector components $u_i(1)$, $i = 1,2,\ldots, m$, which reduces the performance index to ${}^{1}F$ where ${}^{1}F < {}^{o}F$. This new control $\underline{u}(1)$ in general will change the state of the system $\underline{x}(2)$, $\underline{x}(3)$, ..., $\underline{x}(K)$. To calculate $\underline{u}(2)$, for example, we need the new value of

$\underline{x}(2)$ to determine ${}^{*}U_2^j$, $j=1,2,\ldots,m$. It is possible that the set ${}^{*}U_2^j$ does not contain the former "best estimate" for $u_j(2)$ and so the old $u_j(2)$ cannot be selected as an admissible control. This is true because the use of $u_j(2)$ would violate some constraint on $x_i(3)$, or else is not in ${}^{2}U_2^j$. (This follows from the definition of ${}^{*}U_2^j$.) Therefore it is possible that we cannot find $u_j(2) \in {}^{*}U_2^j$ which results in a ${}^{2}F$ such that ${}^{2}F < {}^{1}F$ and in this sense the procedure is not monotonic. The source of difficulty here is large differences in the contents of successive ${}^{*}U_2^j$. These large changes may either arise from discontinuities in the derivatives of the state vector $\underline{x}(k)$ or from large changes in successive values of the input $\underline{u}(1)$. In most physical systems $\dot{\underline{x}}(k)$ is continuous and therefore this generally will not be the source of difficulty. The most frequent cause of the trouble will be successive values of $\underline{u}(1)$ which are considerably different. As observed in the examples of Chapter Four, the first major iteration typically results in a value for the performance index which is well over ninety percent of its optimum value and values for $\underline{u}(k)$, $k = 1, 2, \ldots, K-1$, which are close to their optimum values. Successive changes in $\underline{u}(k)$ are generally very small. Thus successive ${}^{*}U_k^j$ generally differ by very little and thus this non-monotone character of the iteration seldom is observed. This is especially true if G is comparable to F in magnitude so future state space constraints are "strongly" considered as each component of $\underline{u}(k)$ is selected. For the examples studied by the author, no non-monotonic behavior was observed.

If successive passes through all variables produce no further decrease in H, its value at the end of a major iteration, that is $F[\underline{x}(k), \underline{u}(k), K]$, is at a critical point (not a maximum) subject to the constraints on the states and controls. If the function F has saddle point behavior in the m times (K-1) variables $u_i(k)$, $i = 1, 2, \ldots, m$; $k = 1, 2, \ldots, K-1$, the critical point located may not be a minimum. This is clear since the 'one at a time' search procedure cannot distinguish certain saddle points from minima. If the performance index F has no saddle point behavior, the point of inflection located is a local minimum.

Now that the method has been described it should be clear why we are restricted to fixed-time problems. The constraints are fixed in time initially and the system is forced to move through the restricted region of the state space. It is not clear at this time whether or not minimum time problems (for example) could be handled using this technique.

A question which remains to be answered concerns the ability to satisfy the state space constraints at the next instant. It may occur that at a particular instant k, the ${}^{*}U_k^i$ which we have described earlier is empty. This may occur for various reasons. For instance, there may be no solution to the problem (e.g., because of inconsistent specifications). It may be that because the current $\underline{x}(k)$ is not optimum, the ${}^{*}U_k^i$ to be calculated is empty. Work remains to be done to determine whether or not the problem has a solution and, if it does, to develop a technique to account for this particular time instant which seems to have no admissible controls.

4. EXAMPLES

In this chapter the method described earlier is applied to three specific problems. The first problem is deliberately chosen to be simple and it is easy to follow the details of the method. The problem is stated as a discrete one in order to eliminate the necessity for considering the relationship between differential and difference equations. A discussion of such matters may be found elsewhere[19].

The second and third examples are problems described elsewhere but not solved numerically. A detailed discussion of the origin and significance of the problems and their solution are to be found in Sections 4.3 and 4.4.

The examples to be described were programmed on the Control Data Corporation 1604 in the Fortran compiler language. The program is written so any problem of the type considered (i.e., invertible systems) can be run by changing only three subprograms. In these subprograms, the system, the inverted system and the performance index are described. Data to the program includes range of initial conditions, state and control space constraints, final time and step size and a list of parameters which may be used as desired.

A block diagram of the program is shown in Appendix B. The terminology of the flow chart is the same as in Chapter Three.

Several observations about the examples are in order. First, all curves are drawn with continuous lines joining data points. This is to help visualize the results only. Actually data only exists at a discrete set of points which are clearly marked on each sheet.

The trajectory problems show a very strong sensitivity to initial conditions. For this reason, the ability to optimize initial conditions as well as controls is of fundamental importance. In practice it is frequently true that initial conditions may be selected within a certain range. The trajectory examples indicate that substantial advantages exist by careful choice of these initial conditions.

One point of significance is the effect of multipliers on the performance index. In the examples shown in this chapter terminal constraints are handled by means of penalty functions, i.e., positive semi-definite functions of the difference between actual and desired final states. Each terminal constraint is weighted according to its importance relative to fuel cost, thrust limitations, vehicle lifetime, etc. Proper selection of the weighting factors can be of considerable importance in obtaining realistic answers. Much more should be said about these considerations. However, this is not the place for it. It is sufficient to say that a poor choice of weights will preclude a realistic solution of these optimization problems.

4.1 Example One - A Discrete Linear System

Figure 4.1-1 shows a block diagram of the system under consideration. Its state vector representation is given below:

$$\begin{aligned} x_1(k+1) &= -x_1(k) + x_2(k) + u_1(k) \\ x_2(k+1) &= +x_1(k) - x_2(k) + u_2(k) \\ k &= 1, 2, \ldots, 10 \end{aligned} \tag{4.1.1}$$

State and control space constraints shown in Figure 4.1-2 are

$$\begin{aligned} \underline{x}(1) &= \underline{1} \\ -\underline{1} \leq \underline{x}(k) &\leq +\underline{1} \\ -\underline{1} \leq \underline{u}(k) &\leq +\underline{1} \end{aligned} \tag{4.1.2}$$

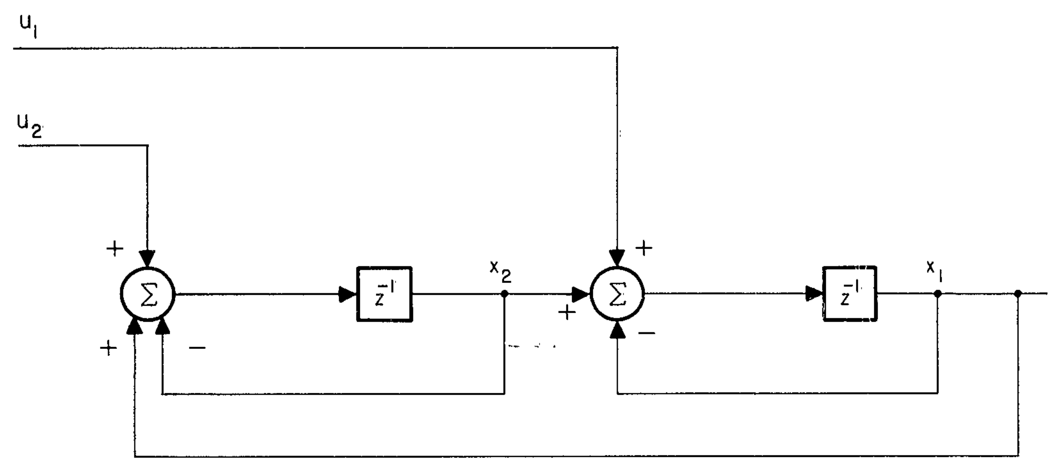


Figure 4.1-1. Discrete System for Example One

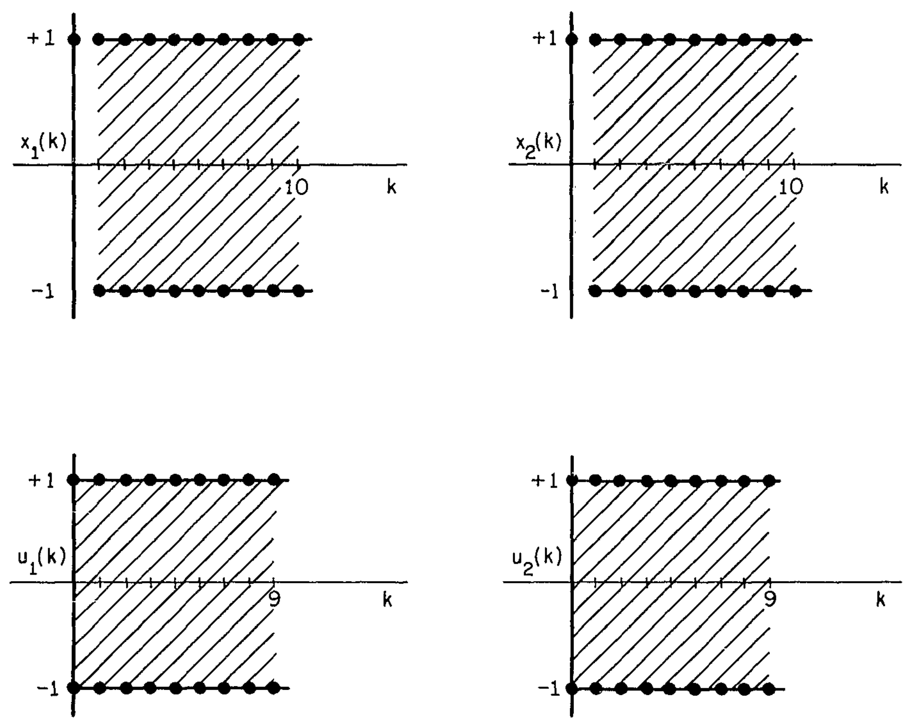


Figure 4.1-2. State and Control Space Constraints for Example One

where $\pm\underline{1}$ represents a two-vector both of whose elements are +1 or -1, respectively.

The performance index chosen for this problem is:

$$F = \sum_{j=1}^{2} \sum_{k=2}^{9} x_j^2(k) + \sum_{j=1}^{2} \sum_{k=1}^{9} u_j^2(k) + M \left| x_1(10) - .5 \right| \qquad (4.1.3)$$

where M is a constant chosen to drive $x_1(10)$ sufficiently close to .5.

Initial conditions selected are $x_1(1) = x_2(1) = +1$. Initially the optimal control is guessed to be $\underline{u}(1) = 0$, $k = 1,2,\ldots,10$.

With M = 1.0, Figures 4.2-3 and 4.2-4 show the state variables at the end of certain major iterations. Figures 4.2-5 and 4.2-6 show the control variables for certain major iterations.

The value of the performance index shown in Figure 4.2-7 is seen to decrease monotonically with the number of major iterations as explained in Chapter Three. If we consider (just for illustration) that its final value is .115, then it is within .03 of its final value after just one iteration. That is, in one iteration it has decreased more than 99.999% of the total decrease to be expected. It might be noted that many numerical procedures converge extremely quickly during the first few iterations, then quite slowly after that. Using the FORTRAN program mentioned above, each (major) iteration required about ten seconds on the CDC 1604.

One other appropriate remark involves the results shown in Figures 4.1-3 and 4.1-4. There at the zeroth iteration we note that state space constraints are violated. This results from certain of the controls being inadmissible. In the terminology of Chapter Three, certain of the "first guess" controls are not in ${}^{**}U_k^j$ although it is seen that they do belong to ${}^{2}U_k^j$. The method assures that this will not occur for any other major iteration.

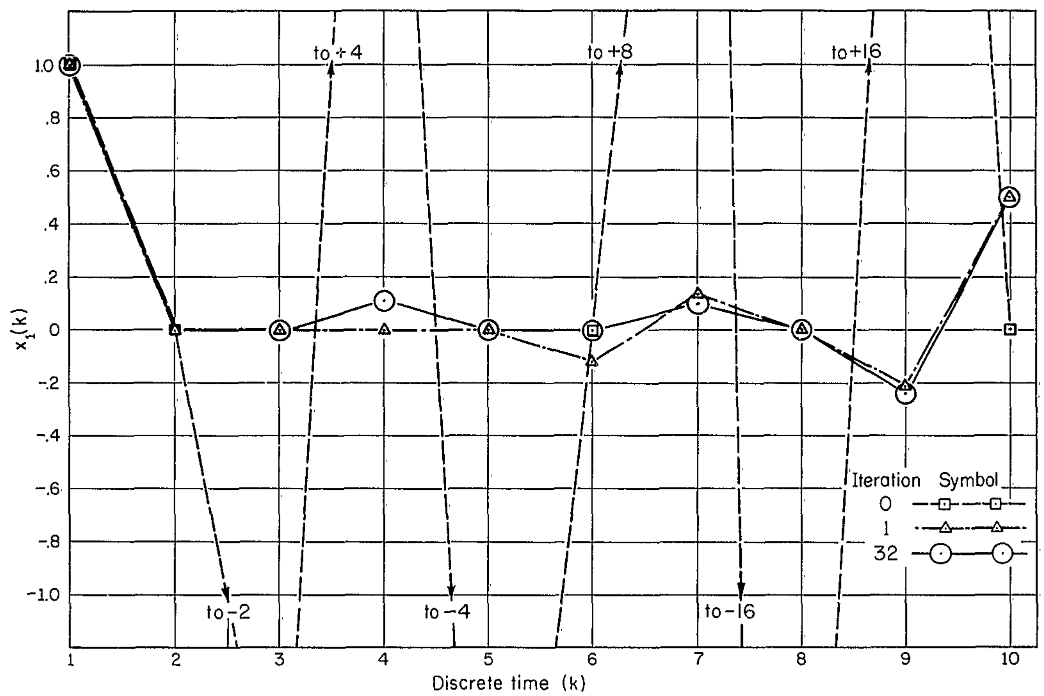


Figure 4.1-3. (Example One) Convergence to Optimal Trajectory $x_1(k)$ vs. Time

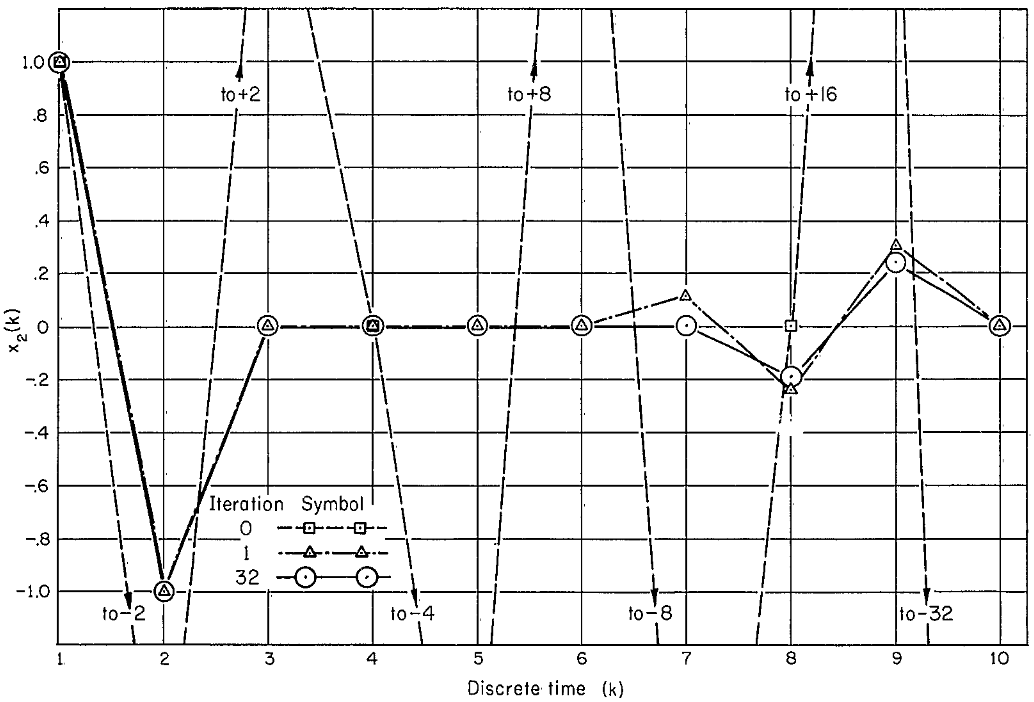


Figure 4.1-3. (Example) Convergence to Optimal Trajectory x_2 (k) vs. Time

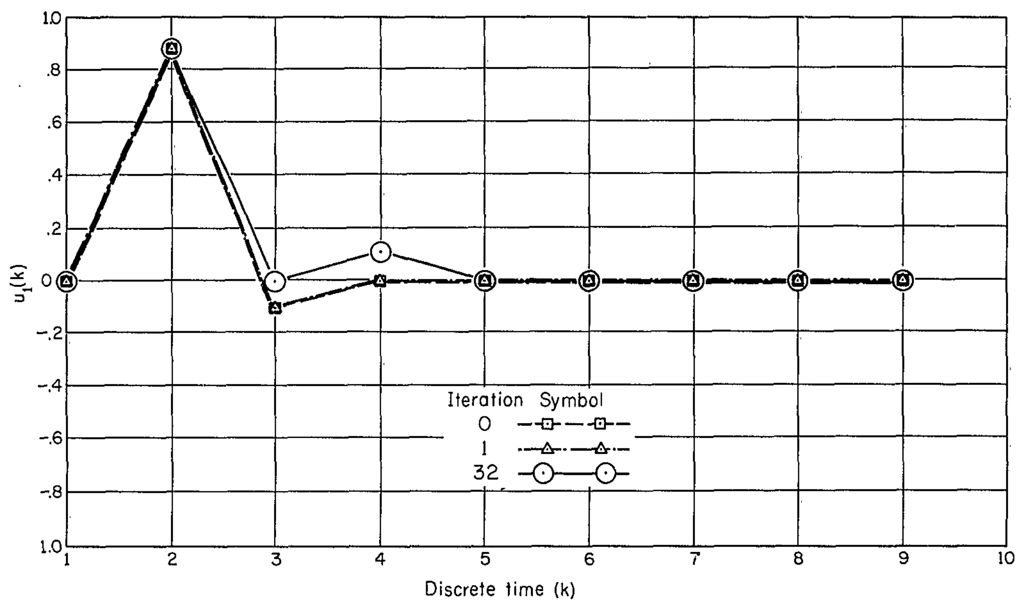


Figure 4.1-5. (Example One) Convergence to Optimal Control $u_1(k)$ vs. Time

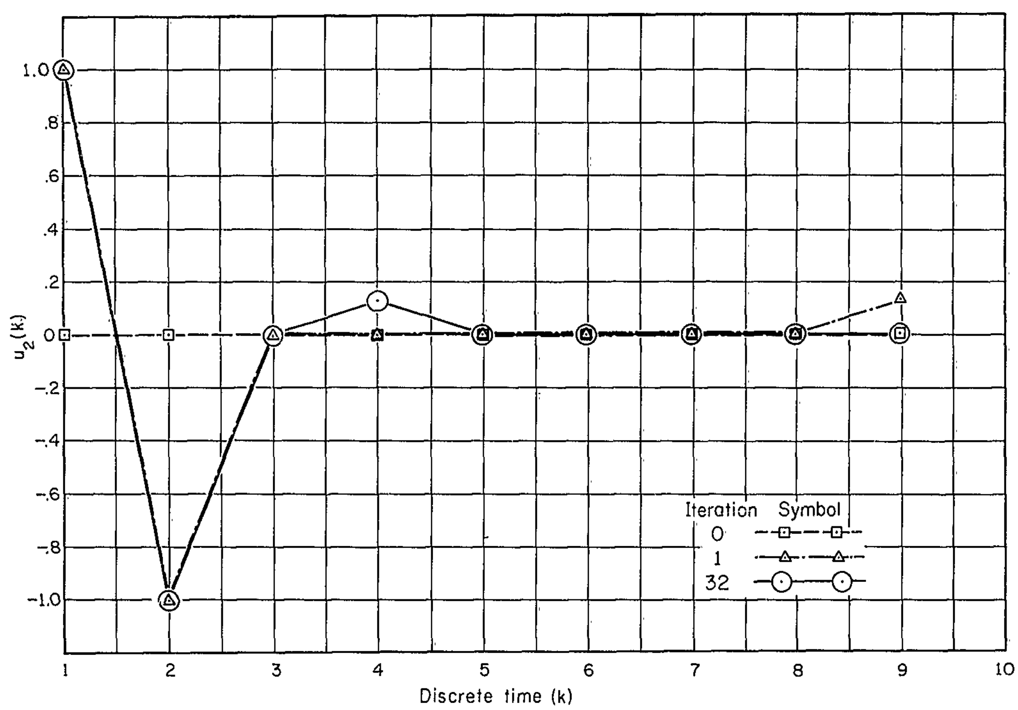


Figure 4.1-6. (Example One) Convergence to Optimal Control $u_2(k)$ vs. Time

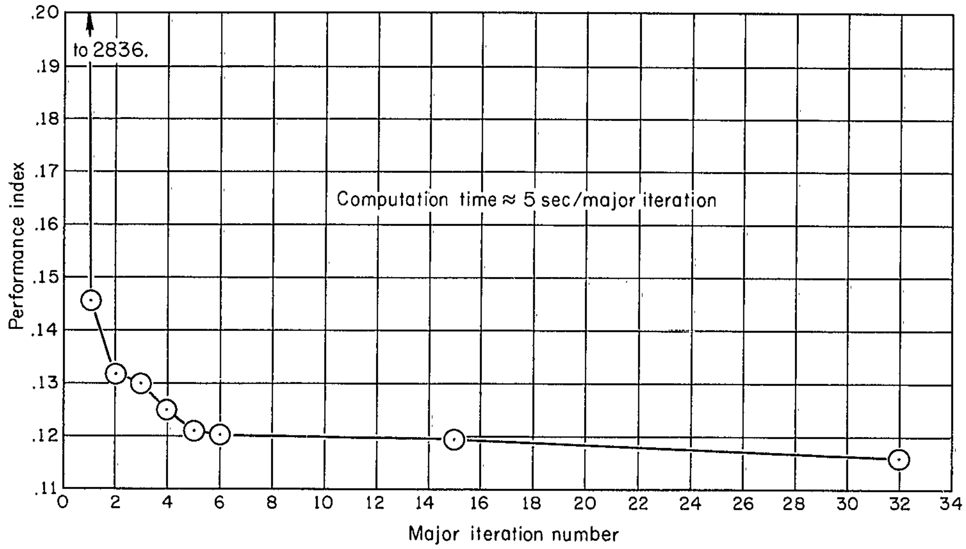


Figure 4.1-7. (Example One) Performance Index vs. Major Iteration Number

4.2 Example Two - A Midcourse Guidance Problem

In this example the problem of interorbit transfer or more commonly, the midcourse guidance problem[20], is examined. Consider the equations of motion of the center of mass of a vehicle moving in two dimensions in the earth's gravitational field. The geometry of this two-dimensional, restricted two-body problem is shown in Figure 4.2-1. The equations follow:

$$\begin{aligned}
\mathring{r}_1 &= r_2 \\
\mathring{r}_2 &= r_1\theta_2^2 - \frac{N}{r_1^2} + \frac{U_1}{m} \\
\mathring{\theta}_1 &= \theta_2 \\
\mathring{\theta}_2 &= -2\,\frac{r_2\theta_2}{r_1} + \frac{U_2}{mr_1}
\end{aligned} \tag{4.2.1}$$

where

$N = GM = 10^{14}\ nm^2/Kg.$

G = universal gravitational constant

M = mass of the earth

m = mass of vehicle = 10^3 Kg.

To use the notation of the other examples we define $r_1 = x_1$, $r_2 = x_2$, $\theta_1 = x_3$, $\theta_2 = x_4$. We will consider that the vehicle is to be placed into an earth orbit which takes it through a certain "launching" region in space. From the optimal point (to be selected) in this "launching" region the vehicle is to be transferred to a specified point in a different orbit in a given length of time. Minimum fuel is to be used and each rocket has a specified maximum thrust.

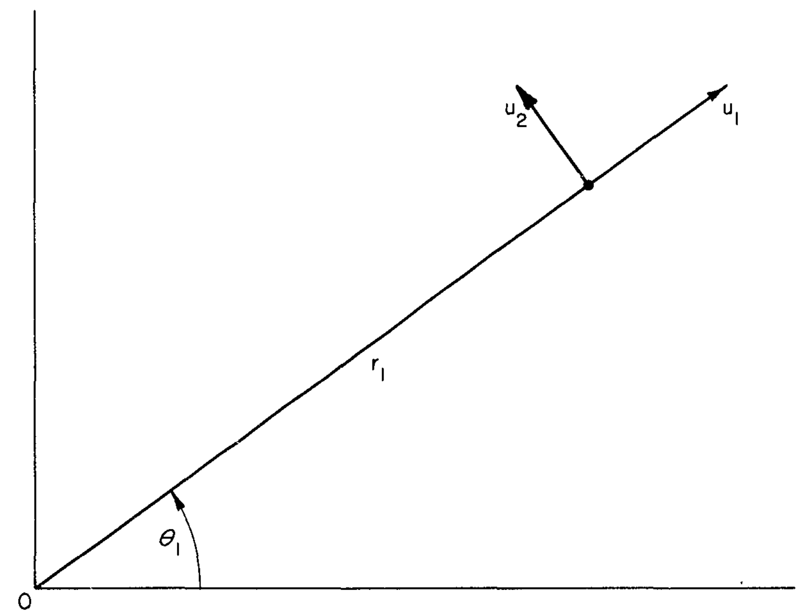


Figure 4.2-1. Geometry for Example Two

The main thruster is constrained to exert force only in the +r direction. The transverse rockets exert force in the $\pm\theta$ directions. We are to determine the optimum "launch" point and control program subject to certain constraints.

Specifically, the constraints are given below.

$$\begin{aligned} 0 \le u_1 \le 500 \\ -200 \le u_2 \le +200 \end{aligned} \tag{4.2.2}$$

We want to minimize $\int_0^{960} (u_1^2(t) + u_2^2(t))\,dt$ where force is in newtons, time in seconds, length in meters, mass in kilograms, and angles in radians. The initial state is given by

$$\underline{x}^-(0) \le \underline{x}(0) \le \underline{x}^+(0)$$

where

$$\underline{x}^-(0) = \begin{bmatrix} 10^7 \\ 0 \\ 0 \\ 0 \end{bmatrix} ; \qquad \underline{x}^+(0) = \begin{bmatrix} 10^7 \\ 0 \\ 1 \\ .005 \end{bmatrix}$$

The final desired state is specified by only two components of the state vector $x_1(960) = 1.5\text{x}10^7$, $x_3(960) = 1.0$ and the other components are free. Figures 4.2-2 and 4.2-3 show the optimum trajectory and Figure 4.2-4 shows the optimum control. (Strictly of course these are only close to the optimal control. This is clear by examining Figure 4.2-5.) For this problem, relatively little control effort is required. The initial conditions which are adjustable have a pronounced effect on the trajectory. With a properly selected initial state, the vehicle requires only little control effort.

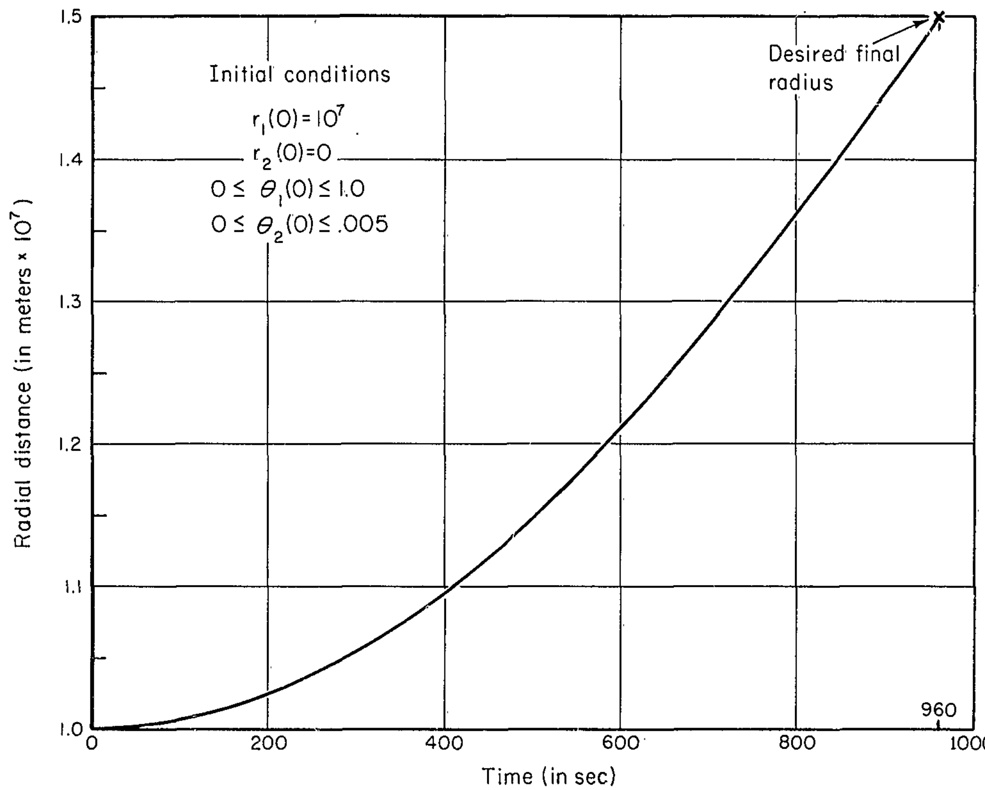


Figure 4.2-2. (Example Two) Radial Distance vs. Time

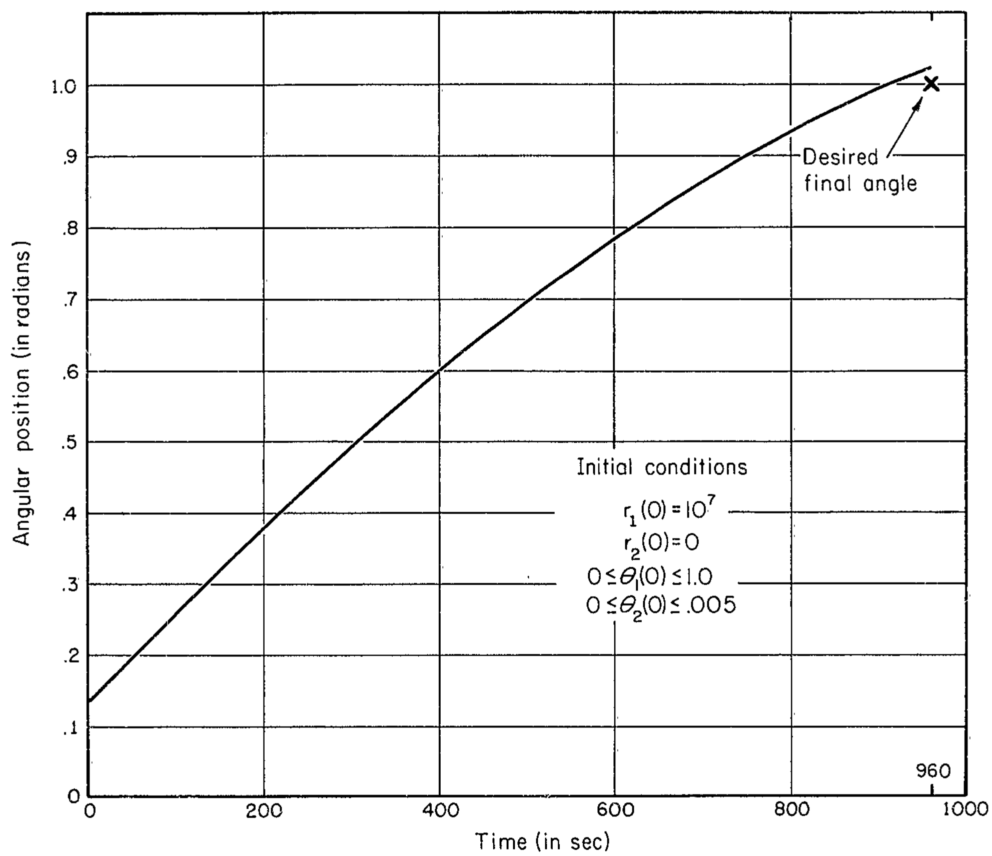


Figure 4.2-3. (Example Two) Angular Position vs. Time

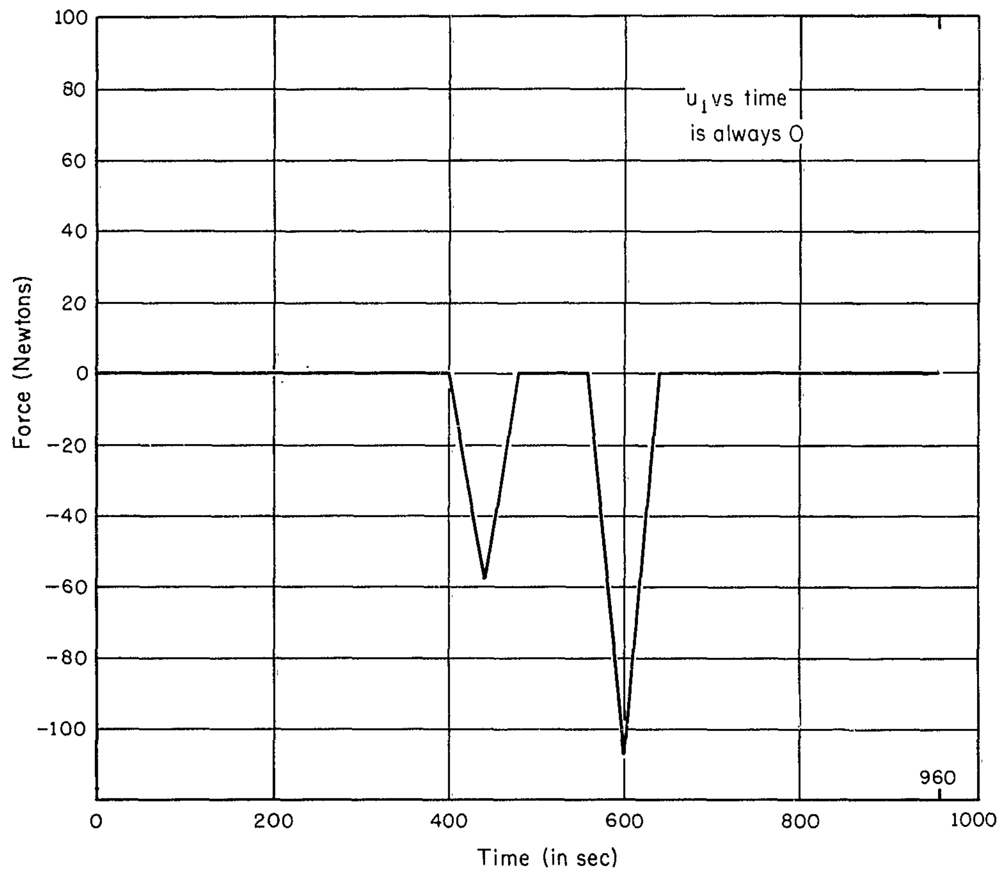


Figure 4.2-4. (Example Two) $u_2(k)$ vs. Time

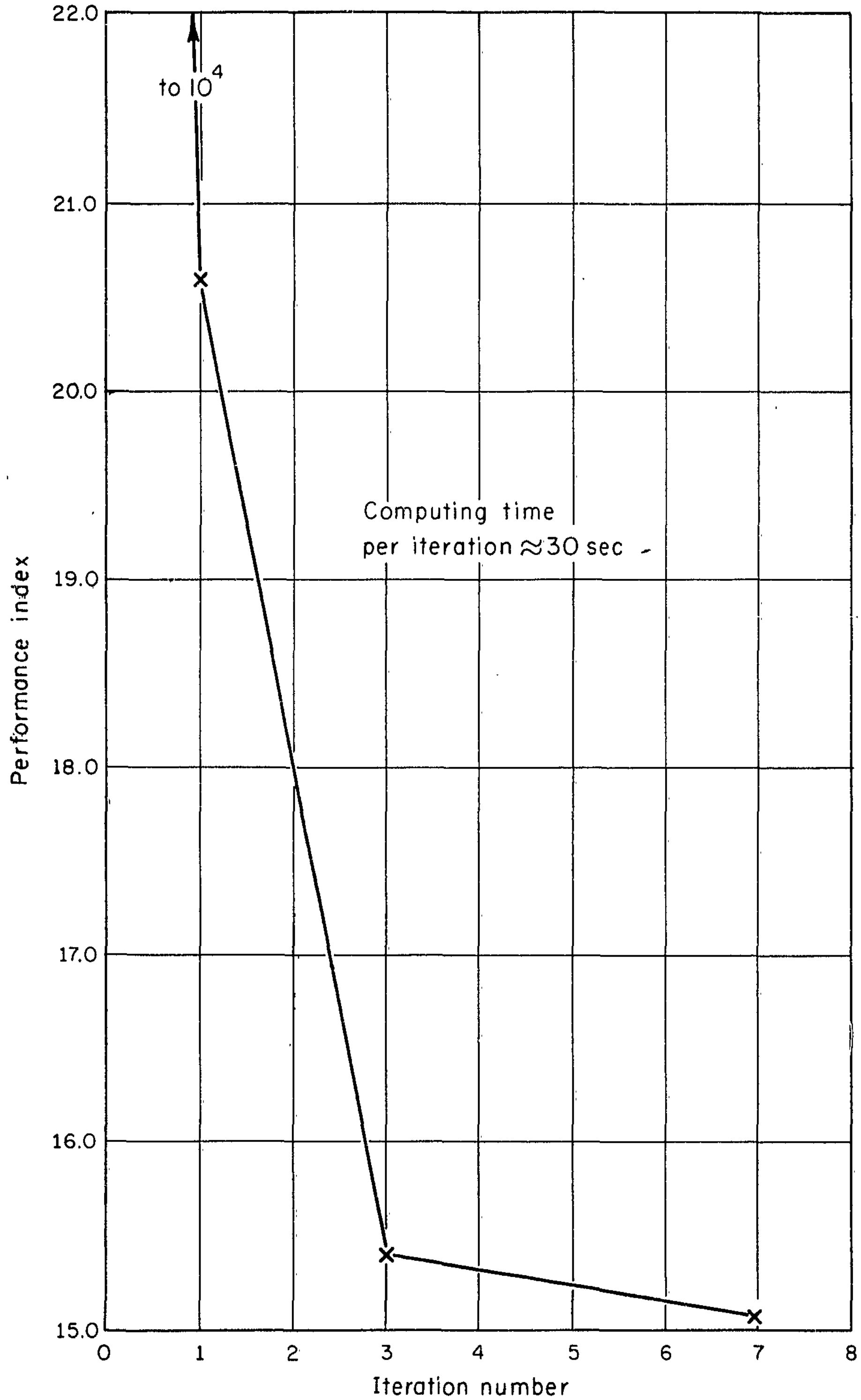


Figure 4.2-5. (Example Two) Performance Index vs. Major Iteration Number

Figure 4.2-5 shows the behavior of the performance index for this problem. This figure exhibits the rapid 'convergence' properties mentioned in connection with Example One.

The performance index used for this problem was

$$F = .001\int_0^{960} (u_1^2(t) + u_2^2(t))\,dt + \left|x_1(960) - 1.5\times10^7\right| + \left|x_3(960) - 1.0\right|$$

4.3 Example Three - A Lunar Landing Problem

The problem of landing manned or unmanned vehicles on the lunar surface is an important one currently being studied. There are several formulations of the problem, the one here being due to Friedland[2]. In Figure 4.3-1 the coordinates r and θ are defined. r is the distance from the moon's center and θ is a measure of angular position. As in the previous example we assume two sets of rockets are mounted on the vehicle, one which exerts force in the $+r$ direction and a smaller pair to exert thrust in the $\pm\theta$ directions. Since the vehicle is assumed to be in the vicinity of the moon when this phase of the flight begins, three basic assumptions are employed in the following derivation. First it is assumed that the total mass of fuel required for the descent is small compared to the mass of the vehicle so the vehicle mass may be assumed constant. The flight time of this phase is small with respect to the moon's period about the earth (about 1 revolution/28 days) so that Coriolis and centrifugal forces arising from rotation of the coordinate system (origin at earth, rotating with moon) may be neglected. Since r/r_o is very small for this phase, terms in r/r_o of first and higher order will be neglected if compared with unity.

g_m, R_m, M_m are the lunar gravitational acceleration, radius and mass, respectively.

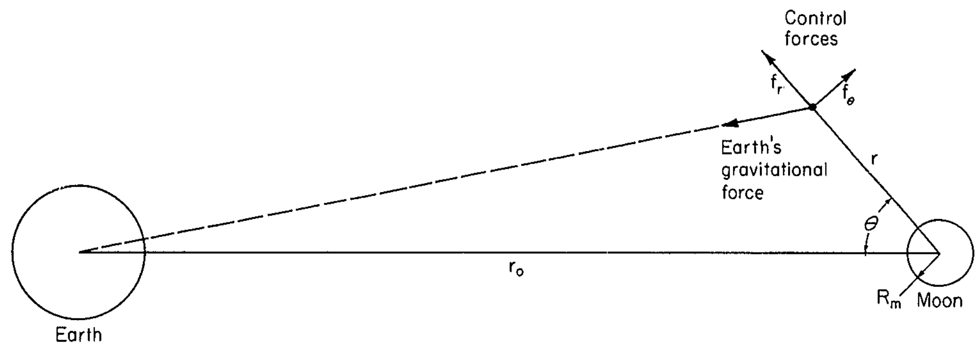


Figure 4.3-1. Geometry for Example Three

g, R, M_e are the earth's gravitational acceleration, radius and mass, respectively.

v_e is the escape velocity from the moon's surface (neglecting the earth's presence).

$$v_e = 2g\left(\frac{M_m \; R^2}{M_e \; R_m}\right)^{1/2}$$

$$T_e = \frac{R_m}{v_e} = 733 \text{ sec.}, \quad \tau = t/T_e$$

$$K = \frac{1}{2}\,\frac{M_e R_m^{\;2}}{M_m r_o^{\;2}} = .0084$$

$$u_1 = \frac{R_m}{v_e^{\;2}\, m}\, f_r = 3.03\, \frac{f_r}{W}$$

$$u_2 = \frac{R_m}{v_e^{\;2}\, m}\, f_\theta = 3.03\, \frac{f_\theta}{W}$$

W = weight of vehicle on earth.

With these definitions and assumptions we can write the equations of motion.

$$\begin{aligned}
\frac{dx_1}{d\tau} &= (1 + x_3)\, x_2^2 - \frac{1}{2(1 + x_3)^2} + K \cos x_4 + u_1 \\
\frac{dx_2}{d\tau} &= \frac{2x_1\, x_2}{1 + x_3} - \frac{K \sin x_4}{1 + x_3} + \frac{u_2}{1 + x_3} \\
\frac{dx_3}{d\tau} &= x_1 \\
\frac{dx_4}{d\tau} &= x_2
\end{aligned} \tag{4.3.1}$$

where

x_1 is a normalized radial velocity

$x_1 = 1$ is the moon's escape velocity

x_2 is a normalized angular velocity

x_3 is a normalized radial distance

$x_3 = 0$ is one moon radius (denoting the lunar surface)

x_4 is exactly the angle θ.

The problem solved here used the following constraints:

$$0 \leq u_1(k) \leq 10.0$$

$$-5.0 \leq u_2(k) \leq +5.0$$

$$x_3(k) \geq 0$$

$$\begin{bmatrix} -0.2 \\ -0.5 \\ +0.2 \\ -2.0 \end{bmatrix} \leq \underline{x}(1) \leq \begin{bmatrix} -0.2 \\ +0.5 \\ +0.2 \\ -2.0 \end{bmatrix}$$

$$\sum_{k=1}^{24} u_1(k) \leq 20.0$$

$$\sum_{k=1}^{24} \left| u_2(k) \right| \leq 50.0$$

$\tau = 0$ corresponds to $k = 1$

$\tau = 2.4\text{x}733$ sec. corresponds to $k = 25$

Time between successive values of k is 73.3 sec. The quantity to be minimized is fuel consumption:

$$\sum_{k=1}^{24} \left| u_1(k) \right| + \left| u_2(k) \right|$$

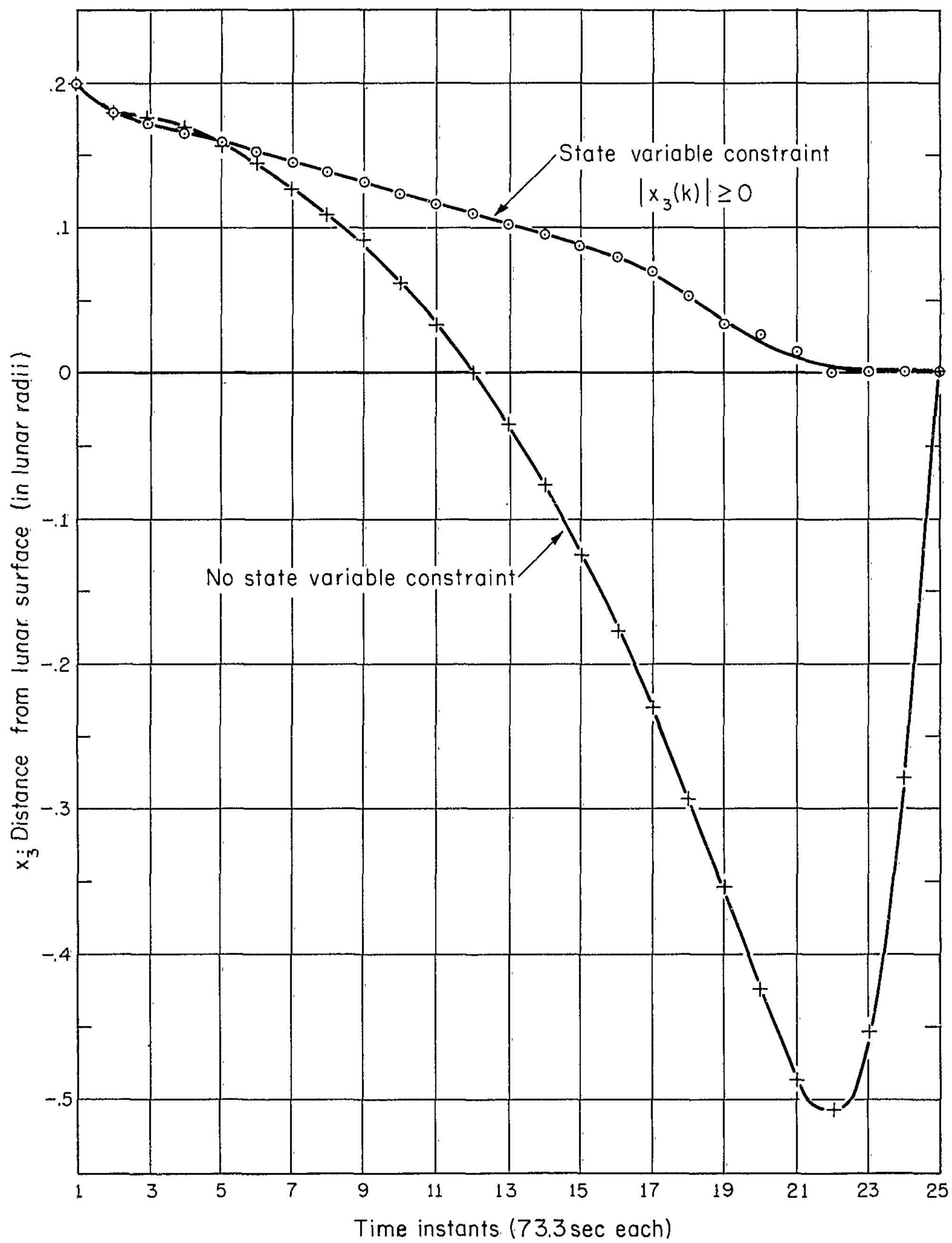


Figure 4.3-2. (Example Three) Radial Distance vs. Time

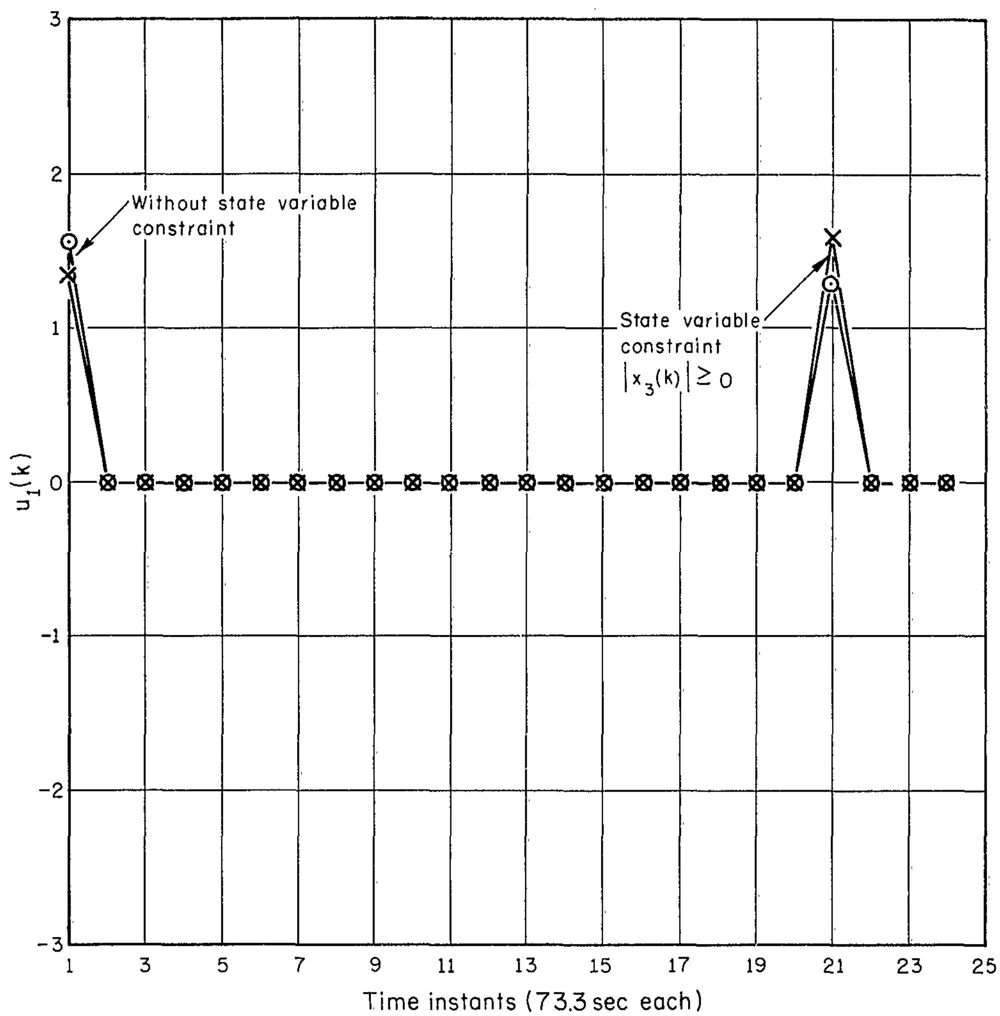


Figure 4.3-3. (Example Three) $u_1(k)$ vs. Time

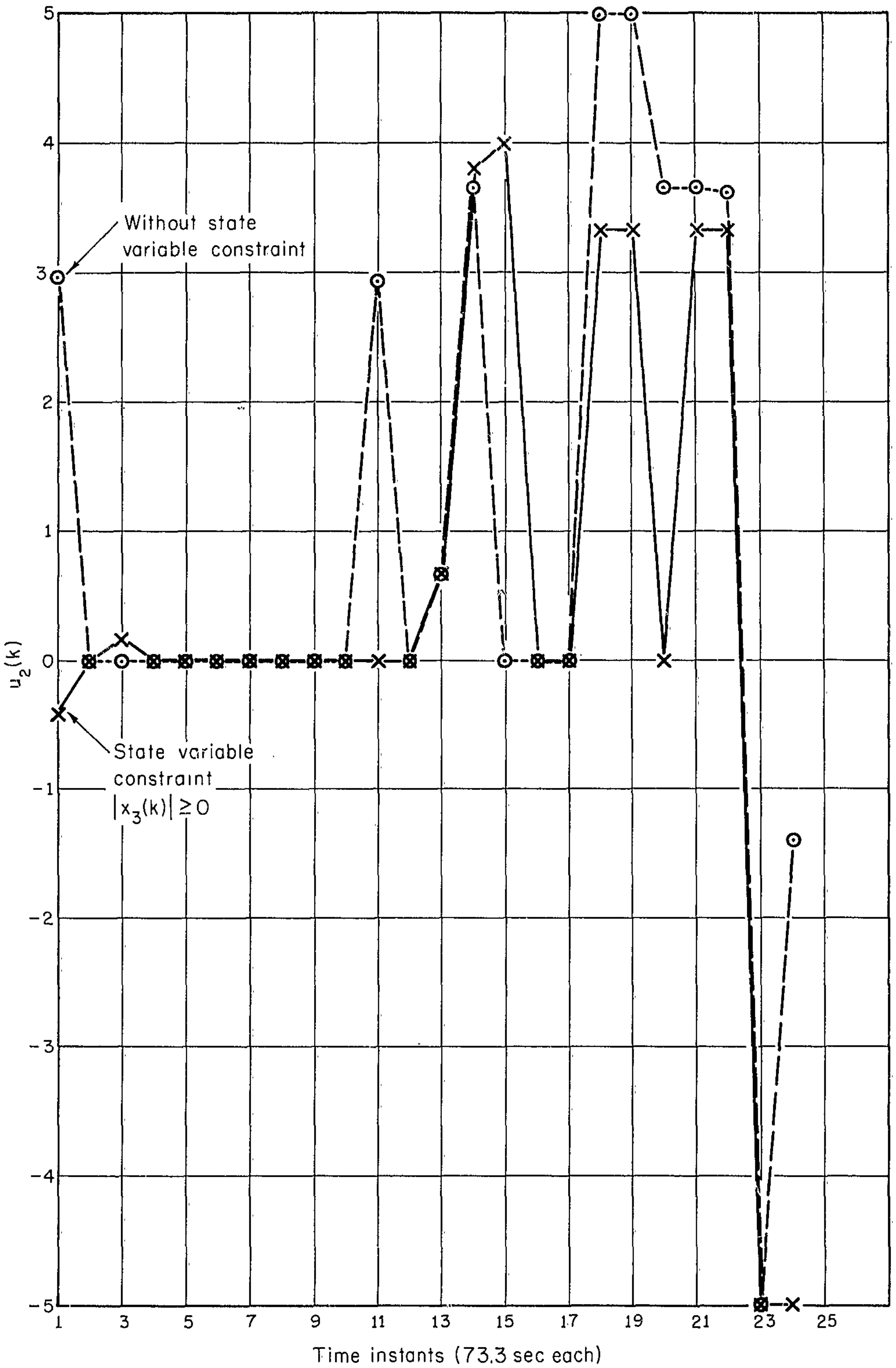


Figure 4.3-4. (Example Three) $u_2(k)$ vs. Time

This is a "difficult" problem because of the large initial value of radial velocity $x_1(0)$. With no control, it was found that the final value of x_1 was -91, a rather unreasonable number.

The curves of Figure 4.3-2 show the result of using a state variable constraint $x_3(k) \geq 0$ to prevent the vehicle from "going under the lunar surface". The control effort required to obtain this trajectory is shown in Figures 4.3-3 and 4.3-4.

Six major iterations of about 45 sec. each were required for each of the two runs.

5. PERSPECTIVE - II

5.1 Summary

A numerical technique for calculating the optimal control for a class of systems and constraints is described. Nonlinear, time-varying deterministic systems subject to hard state space and hard control space constrains are considered. Three numerical procedures are developed to perform the optimization. A technique for the minimization of a scalar function of a vector variable is described where the variables are constrained by upper and lower bounds. This minimization procedure is incorporated in a method of constraint mapping which maps the state space constraints into the control space. To improve convergence properties of the optimization procedure the notion of a pseudo performance index is introduced. Initial and final states may be partially or completely specified. Any unspecified initial or final state vector components are optimally selected.

An iterative technique for the optimization is demonstrated which generally converges to a local minimum of the performance index. The method uses the direct approach to optimization and is very efficient computationally. Examples of space vehicle trajectory optimization problems are given.

5.2 Critique

This thesis is intended as an introduction to a particular approach to totally constrained optimization problems. The systems which can be optimized using the techniques given earlier are quite general. However, certain restrictions were found to be helpful. The notion of invertible systems is, to the author's knowledge, one which has not been explored before. The question naturally arises concerning the necessity or desire for studying invertible systems. It turns out that many common systems currently under study are invertible. Is there something inherent in the structure of invertible systems which makes them of particular interest? Maybe they are not of particular interest. It would seem that further effort might be expended in this direction.

The use of direct optimization methods is not "in style" these days. It does not appear that adequate justification for this state of affairs has been given. The contribution of this thesis is based on a direct approach to optimization problems and the author feels that more effect should be devoted to this area of technology.

To be more specific concerning the results obtained in this thesis, there is one particular difficulty which may arise. Examples can be found to produce the following effect. What explanation might be given if, during the optimization procedure, it was discovered that no control could be found to satisfy state space constraints? If that happened, there are two possible explanations. Either the problem as stated has no solution or else the method has led us astray. We will assume that at time k_i, it is determined that there are no admissible controls. This implies that the state or control space constraints (or both) are "too tight" at this step in the optimization procedure. If we know (from physical reasoning, perhaps) that a solution does exist, we must then "invent" a technique for determining it. A method which could be used is to loosen the constraints for the initial major iterations, then, as the optimum control is approached, to tighten them to their correct vlaues. We are still left with the question of existence if this scheme fails to generate an admissible control. Without quantitative results it would seem that this constraint tightening scheme would eliminate this trouble. However, there is much to be done in this connection. One test which might well be performed is to calculate ${}^{*}U_k^j$ for all j and k. If we find any given ${}^{*}U_k^j = \phi$ then this surely means there is no solution because ${}^{*}U_k^j = \phi$ implies that ${}^{**}U_k^j = \phi$ since ${}^{**}U_k^j \subseteq {}^{*}U_k^j$.

A significant advantage of the method proposed over other computational schemes is shorter computation time. Running times for the three examples were given in Chapter Four. These times are less because of inherent computational advantages of direct methods over indirect ones and also because of the number of "sampling" intervals considered. Just to mention one aspect of the indirect approach, two point boundary value problems are quite costly with respect to computation time. (Details of the two point boundary value problem may be found elsewhere[21].) Using the method of this thesis, many fewer partial derivations are needed since we do not require travel in the direction of steepest descent; many fewer equations need to be integrated because we do not use adjoint systems, etc. On the other hand, the problem of assured convergence mentioned above is not to be underestimated. We can only point out that no work has been published to date which provides a computationally efficient method for solution of the nonlinear time-varying problem with many state and control space constraints. The work of Bryson and Denham[16] is the first effective step in this direction using indirect techniques.

5.3 Comments on Further Engineering Research

Scattered throughout this thesis are ideas and suggestions for further work. In this section, some of these ideas are collected. For this work, we have limited our state space and control space constraints to those of the form of (2.2.3) and (2.2.4) rather than considering $Q[\underline{x}(k), \underline{u}(k), k] \geq 0$. It is true that, in certain cases, the constraints considered here will not be adequate. However, it is also true that when these constraints represent meaningful physical limitations we have demonstrated an optimization technique possessing unusual computational efficiency. We have traded generality for this. This trade is called engineering judgment and such judgment should pervade engineering research.

One area of further study should be to discover what engineering approximations and simplifications can be used to improve existing theoretical results to make them more practical for "real world" applications. The notion of invertibility may be of some importance for future analysis techniques. The method proposed should be extended to non-fixed-time problems. More general constraints may be considered and new results found there.

An engineering constraint which deserves more attention is that of piecewise constant controls in continuous systems. The physical nature of some controlling devices indicates that this is an area of no small importance.

The idea of constraint mapping discussed in these pages may be extended, for example, to performance index mapping in order to permit the actual optimization to take place strictly in the control space. There seems to be much additional work to be done in this field of mapping for optimization.

A feedback solution to the problem posed in this work would be a welcome addition to current research results. There is little work in the literature on this subject.

Bibliography

1. Zadeh, L. A., "From Circuit Theory to System Theory," Proc. of the IRE, Vol. 50, No. 5, pp. 856-865, May 1962.

2. Friedland, B., "Optimum Space Guidance and Control," Technical Note 62/3, Melpar, Inc., June 1962.

3. Kahne, S. J., "A Constraint Mapping Technique for System Optimization," IEEE International Convention Record, Vol. 11, 1963.

4. Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., Mishchenko, E.F., The Mathematical Theory of Optimal Processes, [English translation by Trirogoff and Neustadt], Interscience Publishers, 1962.

5. Chang, S. S. L., "Optimal Control in Bounded Phase Space," TR 400-37, College of Engineering, New York University, August 1961.

6. Breakwell, J. V., "The Optimization of Trajectories," J. Soc. Indust. Appl. Math., Vol. 7, No. 2, pp. 215-247, June 1959.

7. Kelley, H. J., Kopp, R. E. and Moyer, H. G., "Successive Approximation Techniques for Trajectory Optimization," Proc. of the IAS Symposium on Vehicle Systems Optimization, 1961.

8. Bryson, A. E. and Denham, W. F., "A Steepest Ascent Method for Solving Optimum Programming Problems," J. Appl. Mech., pp. 247-257, June 1962.

9. Ho., Y. C., "A Successive Approximation Technique for Optimal Control Systems Subject to Input Saturation," J. of Basic Engineering, Series D, pp. 33-40, March 1962.

10. Ho., Y. C., "A Computational Procedure for Optimal Control Problems with State Variable Constraint," P-2402, The Rand Corp., August 1961.

11. Dreyfus, S., "Variational Problems with Inequality Constraints," J. of Math. Anal. Appl., Vol. 4, No. 2, pp. 297-308, April 1962.

12. Dreyfus, S., "Variational Problems with State Variable Inequality Constraints," P-2605, The Rand Corp., July 1962.

13. Kelley, H. J., Falco, M. and Ball, D. J., "Air Vehicle Trajectory Optimization," presented at SIAM Symposium on Multivariable System Theory, November, 1962.

14. Leitmann, G., (ed), Optimization Techniques, Academic Press, 1962.

15. Ho, Y.C. and Brentani, P. B., "On Computing Optimal Control with Inequality Constraints," presented at 1962 SIAM Symposium on Multivariable System Theory, November, 1962.

16. Bryson, A. E. and Denham, W. F., "The Solution of Optimal Programming Problems with Inequality Constraints," BR-2121, Analytical Research Department, Raytheon Company, November 1962.

17. Stear, E. B., "A Critical Assessment of Aerospace Vehicle Control Techniques," (unpublished) ASD, Air Force Systems Command, WPAFB, Ohio.

18. Kahne, S. J., "The Method of Adaptive Constrained Descent," Report R-154, Coordinated Science Laboratory, University of Illinois, October 1962.

19. Henrici, P., Discrete Variable Methods in Ordinary Differential Equations, Wiley & Sons, New York, 1962.

20. Meditch, J. S. and Neustadt, L. W., "An Application of Optimal Control to Midcourse Guidance," to be published in Proceedings of Second Congress of IFAC, 1963.

21. Kahne, S. J., "Note on Two Point Boundary Value Problems," IEEE Trans. on Automatic Control, July 1963.

22. Speyer, S. L., "Optimization and Control Using Perturbation Theory to Find Neighboring Optimum Paths," BR-2142, Analytical Research Dept., Raytheon Co., December 1962.

23. Dreyfus, S., "The Numerical Solution of Variational Problems," P-2374, The Rand Corp., August 1961.

24. Ahlberg, J. H., Clark, J. W., Williams, J. N., "Optimization by Direct Search Methods with Application to an Adaptive System," Proc. of the IAS Symposium on Vehicle Systems Optimization, 1961.

25. Bliss, G. A., Lectures on the Calculus of Variations, University of Chicago Press, 1946.

26. Berkovitz, L. D., "On Control Problems with Bounded State Variables," J. Math. Anal. Appl., Vol. 5, pp. 488-498, December 1962.

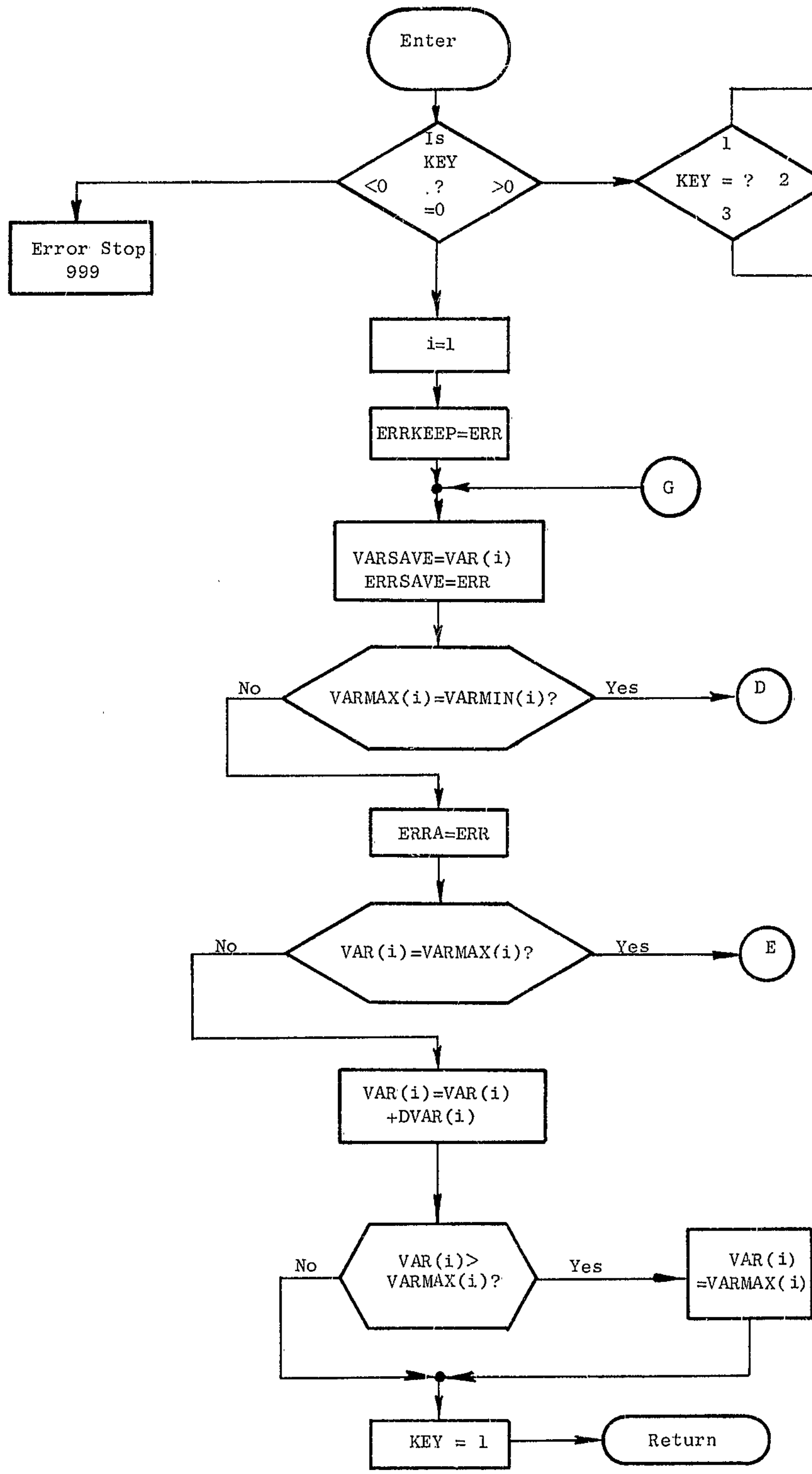
27. Athanassiades, J., et al, "Time Optimal Control for Plants with Numerator Dynamics," IRE Transactions on Automatic Control, Vol. AC-7, No. 4, pp. 47-49, July 1962.

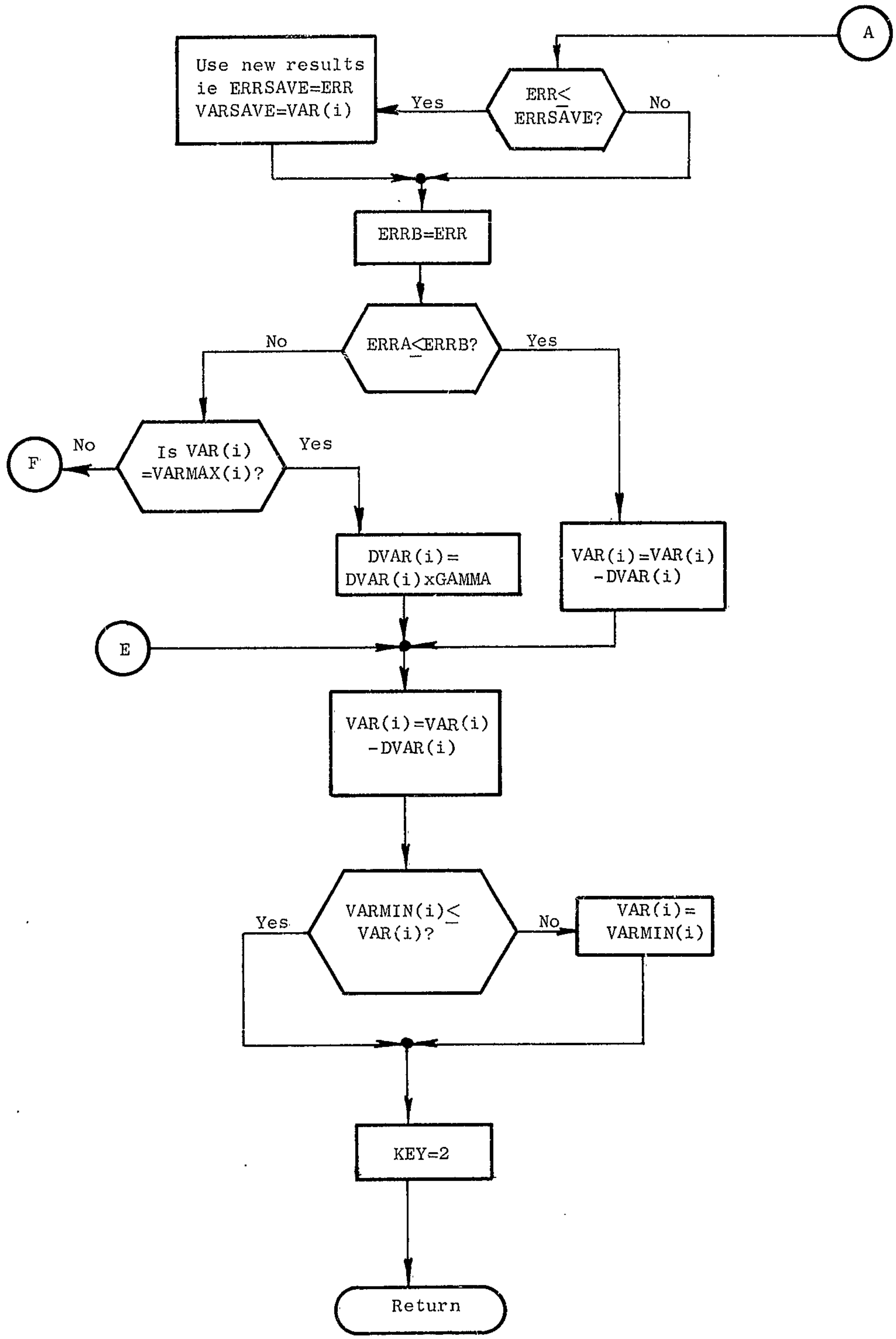
28. Friedland, B., "Techniques for Computation of Optimum Trajectories," Research Report 63-RC-2, General Precision Aerospace, March 1963.

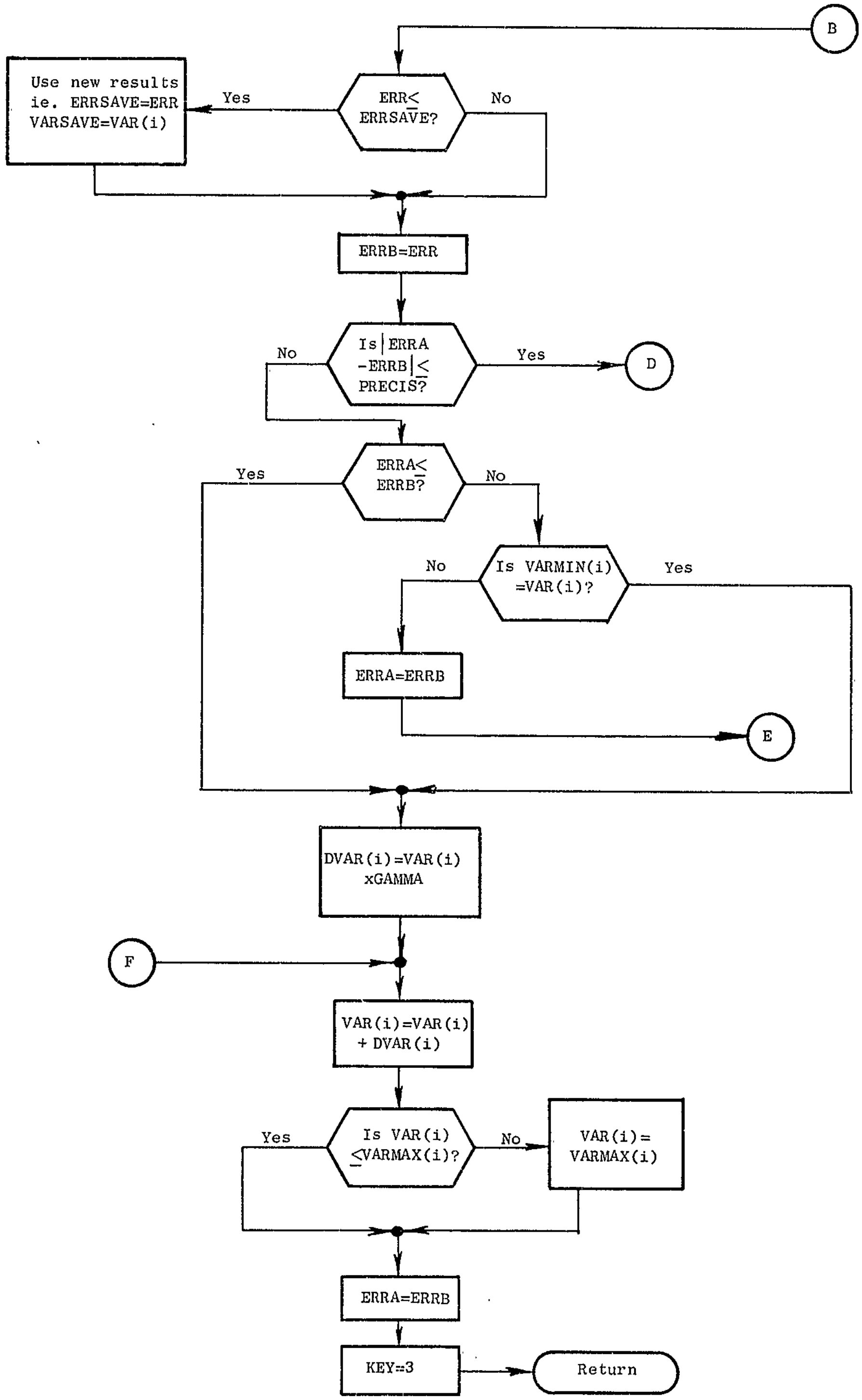
APPENDIX A

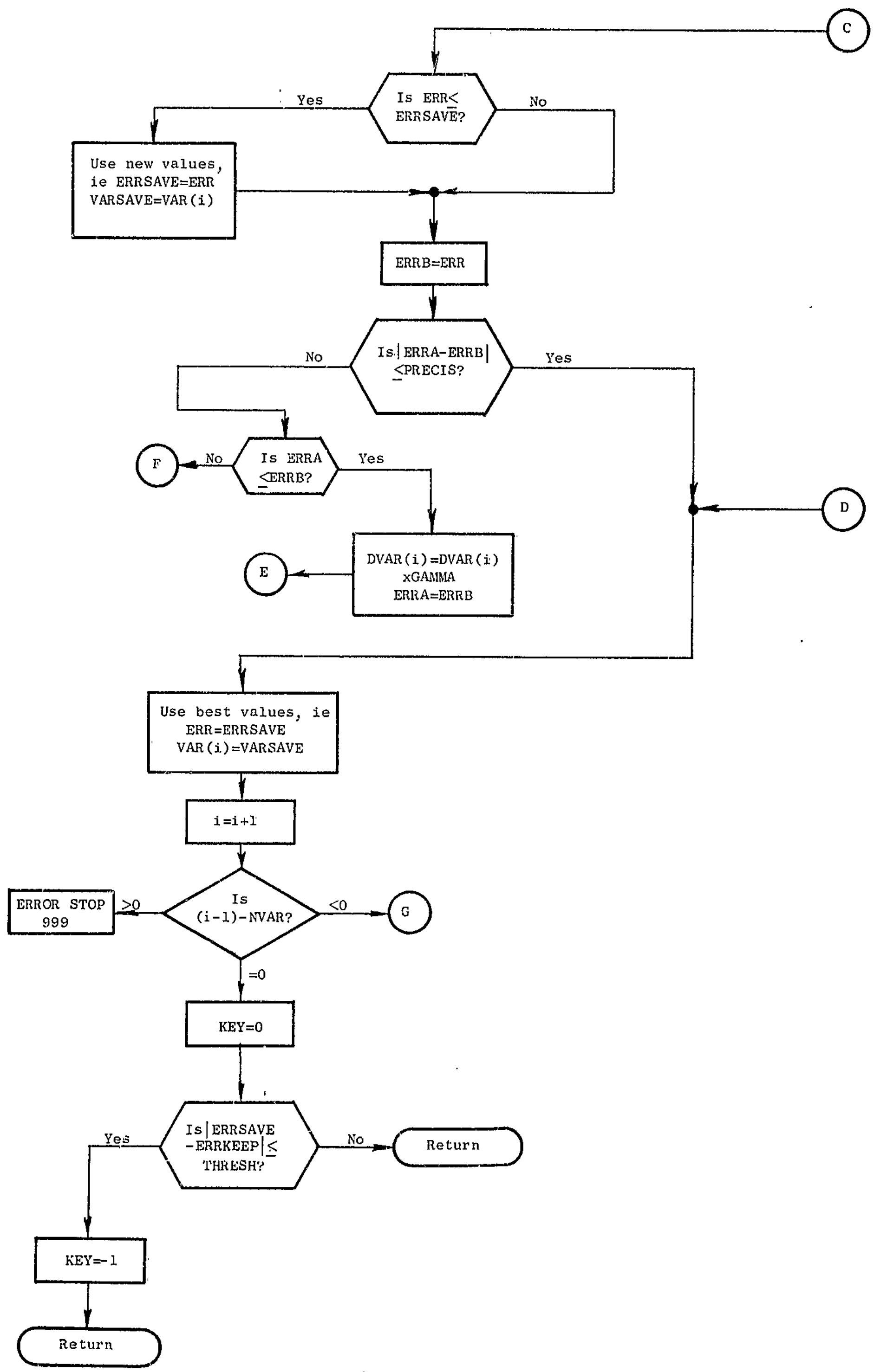
Flow Chart for Adaptive Constrained

Descent Minimization Procedure


Enter
Is KEY ?
<0
>0
=0
Error Stop 999
KEY = ?
1
2
3
A
B
C
i=1
ERRKEEP=ERR
G
VARSAVE=VAR(i)
ERRSAVE=ERR
VARMAX(i)=VARMIN(i)?
No
Yes
D
ERRA=ERR
VAR(i)=VARMAX(i)?
No
Yes
E
VAR(i)=VAR(i)
+DVAR(i)
VAR(i)>
VARMAX(i)?
No
Yes
VAR(i)
=VARMAX(i)
KEY = 1
Return


A
Use new results
ie ERRSAVE=ERR
VARSAVE=VAR(i)
Yes
ERR≤
ERRSAVE?
No
ERRB=ERR
No
ERRA≤ERRB?
Yes
F
No
Is VAR(i)
=VARMAX(i)?
Yes
DVAR(i)=
DVAR(i)xGAMMA
VAR(i)=VAR(i)
-DVAR(i)
E
VAR(i)=VAR(i)
-DVAR(i)
Yes
VARMIN(i)≤
VAR(i)?
No
VAR(i)=
VARMIN(i)
KEY=2
Return

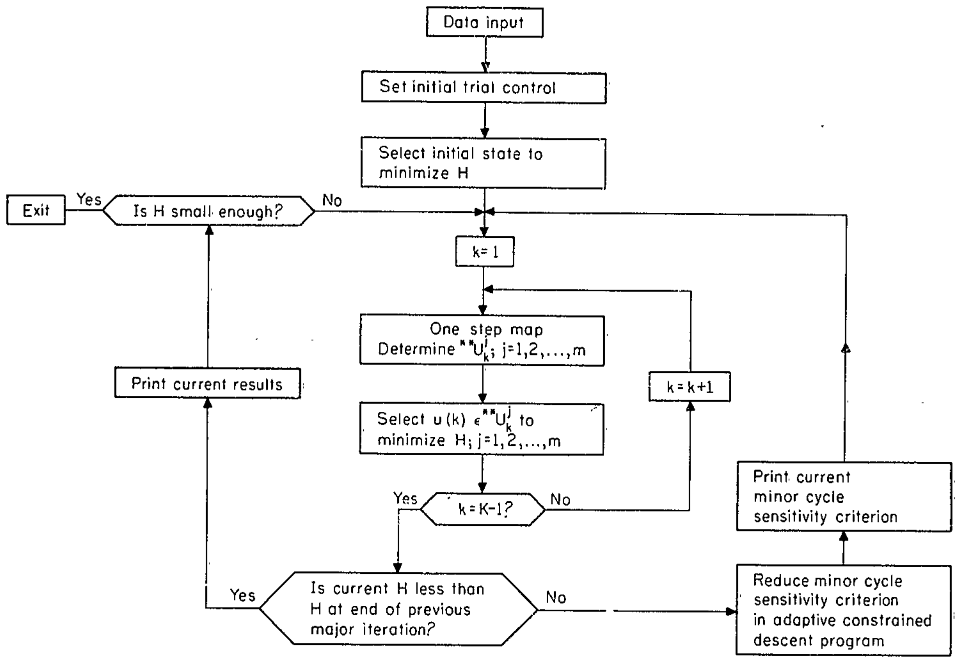

B
ERR≤ ERRSAVE?
Yes
No
Use new results ie. ERRSAVE=ERR VARSAVE=VAR(i)
ERRB=ERR
Is |ERRA -ERRB| ≤ PRECIS?
No
Yes
D
ERRA≤ ERRB?
Yes
No
Is VARMIN(i) =VAR(i)?
No
Yes
ERRA=ERRB
E
DVAR(i)=VAR(i) xGAMMA
F
VAR(i)=VAR(i) + DVAR(i)
Is VAR(i) ≤VARMAX(i)?
Yes
No
VAR(i)= VARMAX(i)
ERRA=ERRB
KEY=3
Return


C
Is ERR≤ ERRSAVE?
Yes
No
Use new values, ie ERRSAVE=ERR VARSAVE=VAR(i)
ERRB=ERR
Is |ERRA-ERRB| ≤PRECIS?
No
Yes
Is ERRA ≤ERRB?
F
No
Yes
D
DVAR(i)=DVAR(i) xGAMMA ERRA=ERRB
E
Use best values, ie ERR=ERRSAVE VAR(i)=VARSAVE
i=i+1
Is (i-1)-NVAR?
ERROR STOP 999
>0
<0
G
=0
KEY=0
Is |ERRSAVE -ERRKEEP| ≤ THRESH?
Yes
No
Return
KEY=-1
Return

APPENDIX B

Flow Chart for Optimization

Program


Data input
Set initial trial control
Select initial state to minimize H
Exit
Yes
Is H small enough?
No
k=1
One step map Determine **U_k^j; j=1,2,...,m
Print current results
k=k+1
Select u(k) ε **U_k^j to minimize H; j=1,2,...,m
Yes
k=K-1?
No
Print current minor cycle sensitivity criterion
Yes
Is current H less than H at end of previous major iteration?
No
Reduce minor cycle sensitivity criterion in adaptive constrained descent program

VITA

Stephen James Kahne was born on 5 April 1937 in New York, New York. In June 1960 he received the BEE degree from Cornell University and in June 1961 the MS from the University of Illinois. As an undergraduate he was employed by the Cornell Aeronautical Laboratory and the General Electric Company. He was a teaching assistant in the School of Electrical Engineering, Cornell University. At the University of Illinois he has been Research Assistant and Research Associate in the Coordinated Science Laboratory and Instructor in the Department of Electrical Engineering. He has published book reviews of two books, "Stability by Lyapunov's Direct Method," and "The Mathematical Theory of Optimal Control." Published papers include. "A Constraint Mapping Technique for System Optimization" and "Note on Two Point Boundary Value Problems."

He is a student member of the Institute of Electrical and Electronics Engineers, associate member of the American Association of University Professors and a member of the Society for Industrial and Applied Mathematics.

ERRATA

1. Pg. 8, Eq. (2.1.5) should be:

$$\underline{\dot{x}}(k+\Delta) = \underline{f}[\underline{x}(k),\Delta] + B[\underline{x}(k),\Delta]\, H[\underline{u}(k)] \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} ;\ k=1,2,\ldots,K$$

2. Pg. 14, first sentence after description of sets should be: It is clear that ${}^{1}U$, ${}^{*}U$ and ${}^{**}U$ are compact.

3. Pg. 16, Eq. (3.2.5) should be:

$$\underline{x}(k+\Delta) = \underline{f}[\underline{x}(k),\Delta] + B[\underline{x}(k),\Delta]\, H[\underline{u}(k)] \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

4. Pg. 21, line 3: ---- k = 1,2,....,K-1. Moreover, ----

5. Pg. 21, line 7: ---- and K-1 minor ----

6. Pg. 22, line 17: ---- and K-1 minor ----

7. Pg. 22, line 18: ---- k=1,2,....,K-1.----

8. Pg. 29, 7th line should be: control is guessed to be $\underline{u}(k) = \underline{0}$, k=1,2,...,9.

9. Pg. 51, line 6: "effect" should be "effort".

DISTRIBUTION LIST AS OF NOVEMBER 7, 1962

1	Director Air University Library Maxwell Air Force Base, Alabama Attn: CR-4803a
1	Redstone Scientific Information Center U.S. Army Missile Command Redstone Arsenal, Alabama
2	Hughes Aircraft Company Florence and Teale Culver City, California Attn: N.E. Devereux Technical Document Center
3	Autonetics 9150 East Imperial Highway Downey, California Attn: Tech. Library, 3041-11
1	Dr. Arnold T. Nordsieck General Motors Corporation Defense Research Laboratories 6767 Hollister Avenue Goleta, California
1	Inspector of Naval Material Los Angeles, California Transmittal to: Litton Systems, Inc. 5500 Canoga Avenue Woodland Hills, California Attn: Engineering Library
1	Sylvania Electronic Systems - West Electronic Defense Laboratories P. O. Box 205 Mountain View, California Attn: Documents Center
1	Varian Associates 611 Hansen Way Palo Alto, California Attn: Dr. Ira Weissman
1	Huston Denslow Library Supervisor Jet Propulsion Laboratory California Institute of Technology Pasadena, California
1	Space Technology Labs, Inc. One Space Park Redondo Beach, California Attn: Acquisitions Group STL Technical Library
2	Commanding Officer and Director U.S. Naval Electronics Laboratory San Diego 52, California Attn: Code 2800, C.S. Manning
1	Commanding Officer and Director U.S. Navy Electronics Laboratory San Diego 52, California Attn: Library
1	Office of Naval Research Branch Office 1000 Geary Street San Francisco, California
1	Stanford Electronics Laboratories Stanford University Stanford, California Attn: SEL Documents Librarian
1	AFRST - SC Headquarters, USAF Washington 25, D. C.
1	Director of Science and Technology Headquarters, USAF Washington 25, D. C. Attn: AFRST-EL/GU
1	Headquarters, R & T Division Bolling Air Force Base Washington 25, D. C. Attn: RTHR
1	Headquarters, U.S. Army Materiel Command Research Division, R & D Directorate Washington 25, D. C. Attn: Physics & Electronics Branch Electronics Section
1	Commanding Officer Diamond Ordnance Fuze Laboratories Washington 25, D. C. Attn: Librarian, Room 211, Bldg. 92
1	Operations Evaluation Group Office of the CNO (Op03EG) Navy Department Washington 25, D.C.
1	Chief of Naval Operations Tech. Analysis & Advisory Group (OP-07T) Pentagon Washington 25, D. C.
1	Commanding Officer & Director David W. Taylor Model Basin Navy Department Washington 7, D. C. Attn: Code 142, Library
1	Bureau of Ships Department of the Navy Washington 25, D. C. Attn: Code 686
1	Bureau of Ships Navy Department Washington 25, D.C. Attn: Code 732
1	Technical Library, DLI-3 Bureau of Naval Weapons Department of the Navy Washington 25, D. C.
1	Director Naval Research Laboratory Washington 25, D.C. Attn: Code 5140
1	Department of the Navy Office of Naval Research Washington 25, D.C. Attn: Code 437
1	Dr. H. Wallace Sinaiko Institute for Defense Analyses Research & Engineering Support Division 1666 Connecticut Ave., N.W. Washington 9, D.C.
1	Data Processing Systems Division National Bureau of Standards Conn. at Van Ness Room 239, Bldg. 10 Washington 25, D.C. Attn: A.K. Smilow
1	Exchange and Gift Division The Library of Congress Washington 25, D.C.
1	NASA Headquarters Office of Applications 400 Maryland Avenue, S.W. Washington 25, D.C. Attn: Mr. A. M. Greg Andrus Code FC
1	APGC (PGAPI) Eglin Air Force Base Florida
1	Martin Company P.O. Box 5837 Orlando, Florida Attn: Engineering Library MP-30
1	Commanding Officer Office of Naval Research, Chicago Branch John Crerar Library Building 10th Floor, 86 East Randolph Street Chicago 1, Illinois
1	Librarian School of Electrical Engineering Purdue University Lafayette, Indiana
2	Keats A. Pullen, Jr. Ballistic Research Laboratories Aberdeen Proving Ground, Maryland
1	Research Analysis Corporation 6935 Arlington Road Bethesda 14, Maryland Attn: Library
5	Scientific & Technical Information Facility P. O. Box 5700 Bethesda, Maryland Attn: NASA Representative (S-AK/DL)
1	Commander Air Force Cambridge Research Laboratories Laurence G. Hanscom Field Bedford, Massachusetts Attn: CRXL
1	Research Laboratory of Electronics Massachusetts Institute of Technology Cambridge 39, Massachusetts Attn: Document Room, 26-327
1	Lincoln Laboratory Massachusetts Institute of Technology P.O. Box 73 Lexington 73, Massachusetts Attn: Library, A-082
1	Sylvania Electric Products Inc. Electronic Systems Waltham Labs. Library 100 First Avenue Waltham 54, Massachusetts
1	Minneapolis-Honeywell Regulator Co. Aeronautical Division 2600 Ridgeway Road Minneapolis 13, Minnesota Attn: Mr. D. F. Elwell Main Station: 625
1	Inspector of Naval Material Bureau of Ships Technical Representative 1902 West Minnehaha Avenue St. Paul 4, Minnesota
20	Activity Supply Officer Building 2504 Charles Wood Area Fort Monmouth, New Jersey Marked: For Office of Engineering Operations Order No. 39578-PM-60-91-91
1	Radio Corporation of America RCA Laboratories David Sarnoff Research Center Princeton, New Jersey Attn: Library
1	Dr. J. H. Frank 861 Washington Avenue Westwood, New Jersey
1	Mr. A. A. Lundstrom Bell Telephone Laboratories Room 2E-127 Whippany Road Whippany, New Jersey
1	AFMDC (MDSGP/Capt. Wright) Holloman Air Force Base New Mexico
1	Mailing Zone C-57 Bell Aerosystems Company P. O. Box 1 Buffalo 5, New York Attn: Technical Library
1	Cornell Aeronautical Laboratory, Inc. 4455 Genesee Street Buffalo 21, New York Attn: J. P. Desmond, Librarian
1	Sperry Gyroscope Company Marine Division Library 155 Glen Cove Road Carle Place, L.I., New York Attn: Mrs. Barbara Judd
1	Rome Air Development Center Griffiss Air Force Base, New York Attn: Documents Library RAALD
1	General Electric Company Advanced Electronics Center Cornell University Ithaca, New York Attn: F. Kuehn, Librarian
1	Library Light Military Electronics Department General Electric Company Building No. 28-501 Schenectady 5, New York
3	Commanding Officer U.S. Army Research Office (Durham) Attn: CRD-AA-IP, Mr. Ulsh Box CM, Duke Station Durham, North Carolina
1	Goodyear Aircraft Corporation For: Project MX 778 Akron 15, Ohio
1	Battelle-DEFENDER Battelle Memorial Institute 505 King Avenue Columbus 1, Ohio
1	ASD (ASNC) Wright-Patterson Air Force Base Ohio
1	ASD (ASRNG) Wright-Patterson Air Force Base Ohio
1	Commanding Officer (AD-5) U.S. Naval Air Development Center Johnsville, Pennsylvania Attn: NADC Library
2	Commanding Officer Frankford Arsenal Philadelphia 37, Pennsylvania Attn: SMUFA-1300
1	General Atronics Corporation 1075 DeHaven Street West Conshohocken, Pennsylvania Attn: Miss D. M. Keener Librarian
1	H. E. Cochran Oak Ridge National Laboratory P. O. Box X Oak Ridge, Tennessee
1	President U.S. Army Air Defense Board Fort Bliss, Texas
1	U.S. Air Force Security Service San Antonio, Texas Attn: ODC-R
1	ASTIA Technical Library AFL 2824 Arlington Hall Station Arlington 12, Virginia Attn: TISLL
1	U.S. Naval Weapons Laboratory Computation and Analysis Laboratory Dahlgren, Virginia Attn: Mr. Ralph A. Niemann
2	Army Materiel Command Research Division R & D Directorate Bldg. T-7 Gravelley Point, Virginia